20STh0"h6Z9E00T

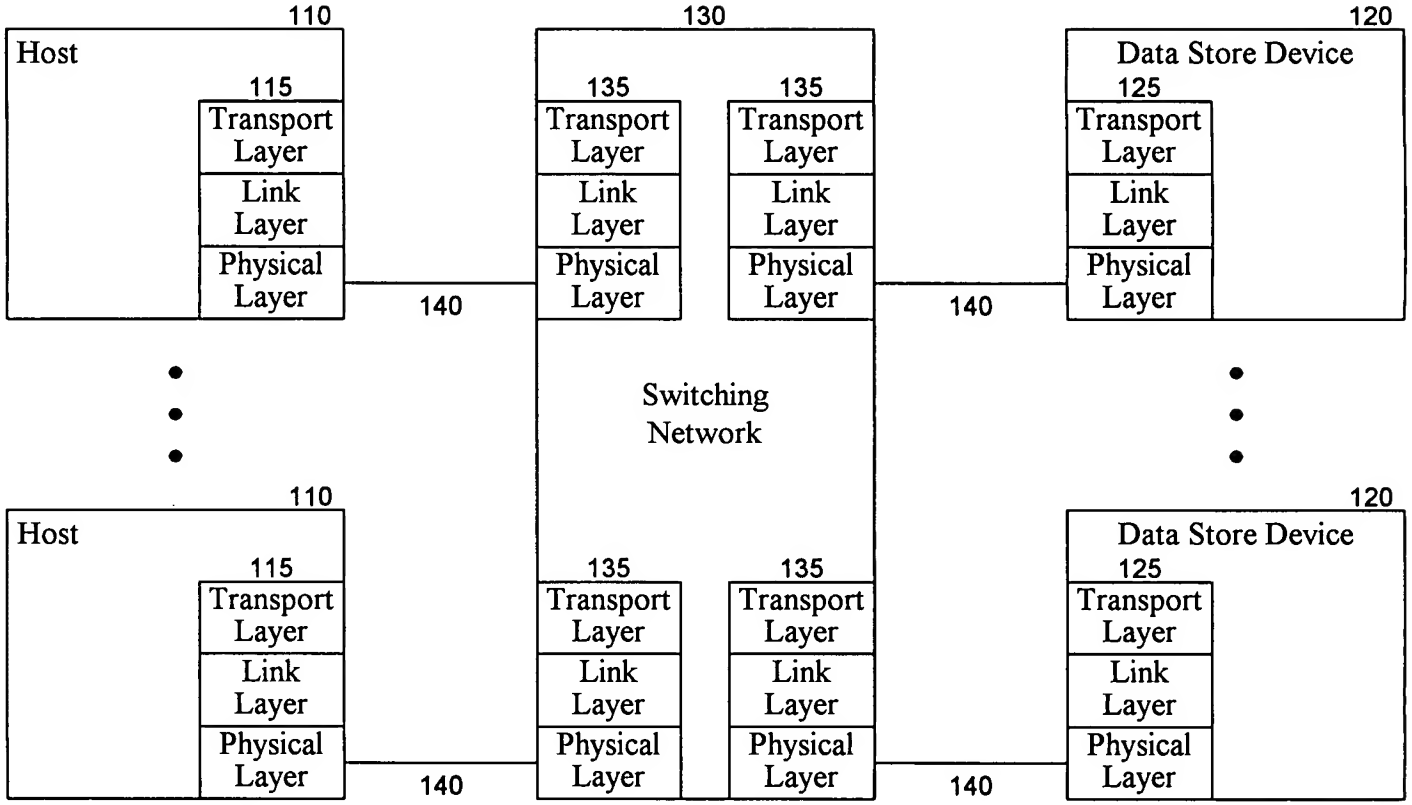



*Fig. 1*

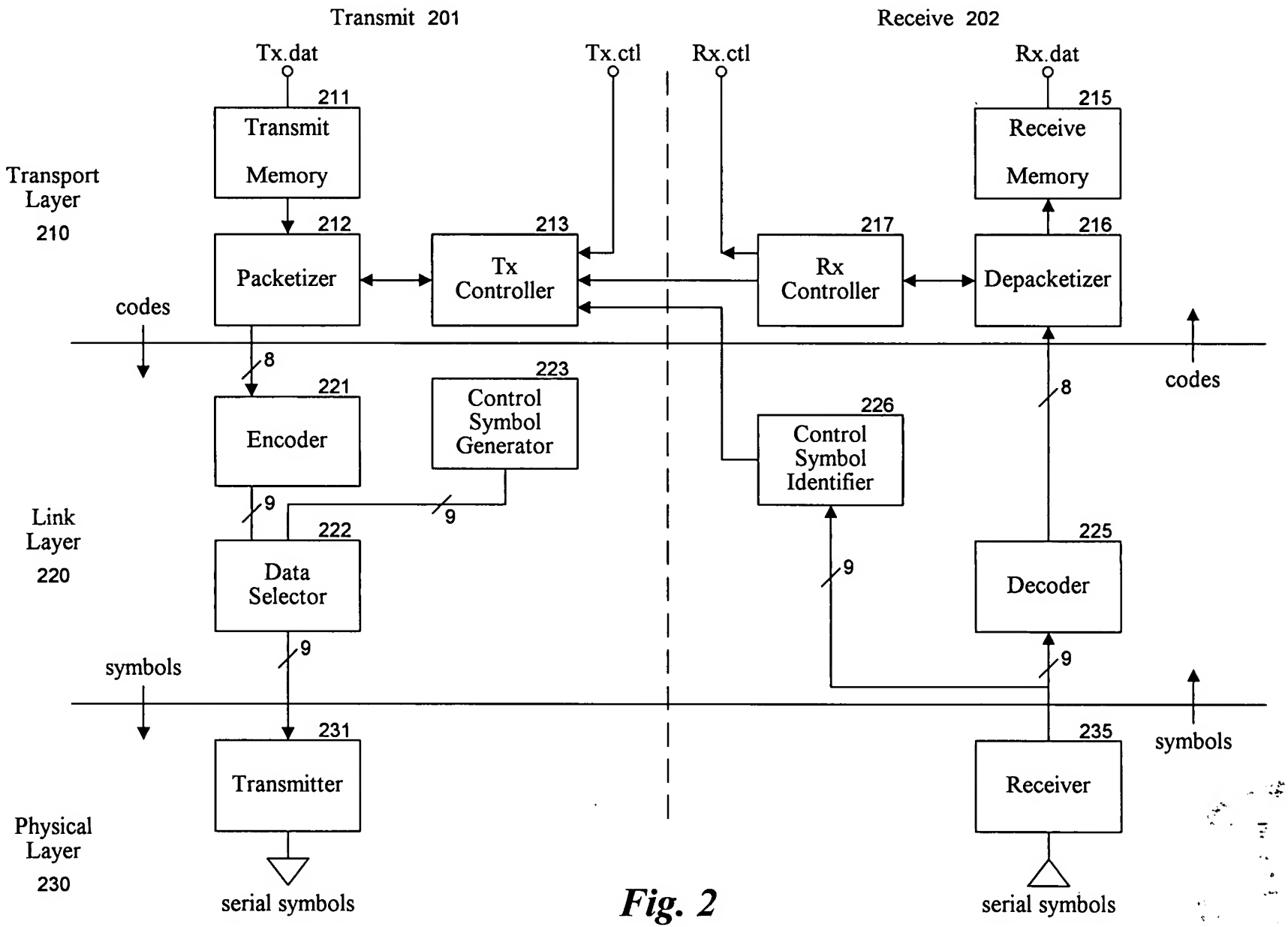

Transmit 201
Receive 202
Tx.dat
Tx.ctl
Rx.ctl
Rx.dat
211
Transmit Memory
215
Receive Memory
Transport Layer 210
212
Packetizer
213
Tx Controller
217
Rx Controller
216
Depacketizer
codes
codes
8
8
223
Control Symbol Generator
221
Encoder
226
Control Symbol Identifier
9
9
9
9
Link Layer 220
222
Data Selector
225
Decoder
symbols
symbols
231
Transmitter
235
Receiver
Physical Layer 230
serial symbols
serial symbols

*Fig. 2*

20STh0" h629E00T

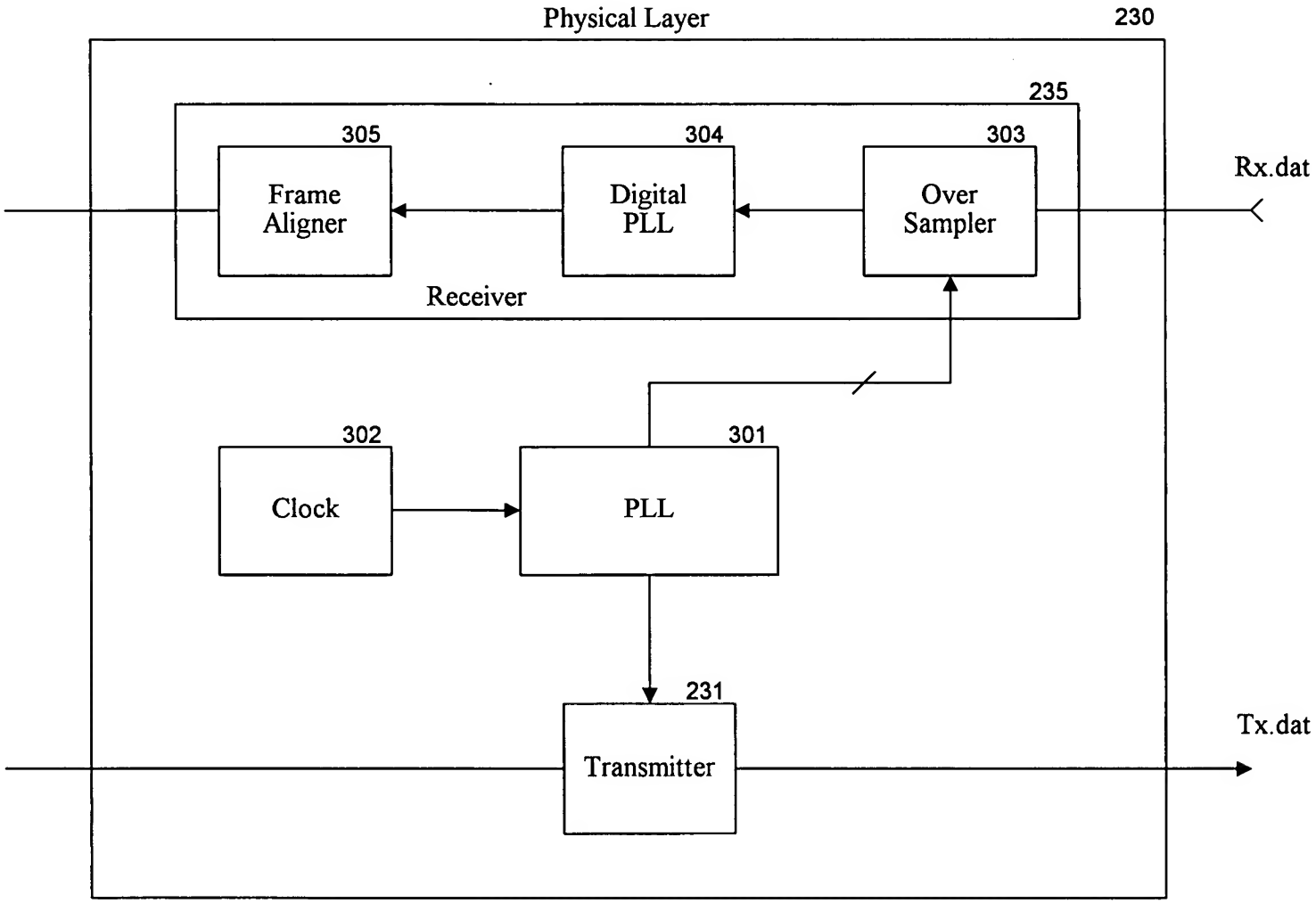



*Fig. 3*

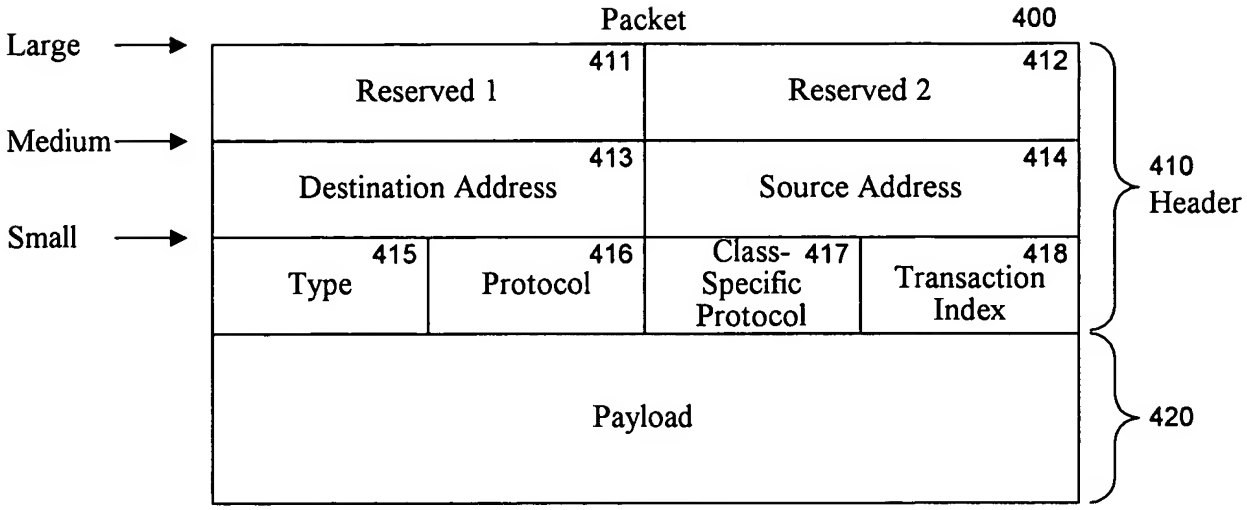

Packet
400
Large
Medium
Small
411
Reserved 1
412
Reserved 2
413
Destination Address
414
Source Address
415
Type
416
Protocol
417
Class-Specific Protocol
418
Transaction Index
410
Header
Payload
420

*Fig. 4*

20STH0 "H629E00T

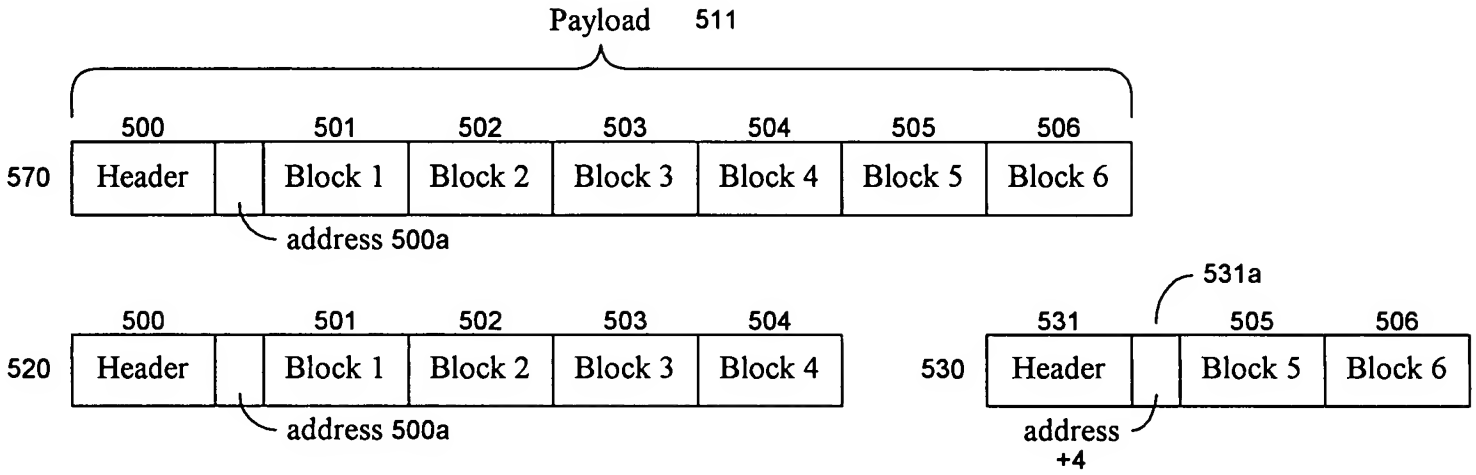



*Fig. 5*

20STh0"h6Z9E0T

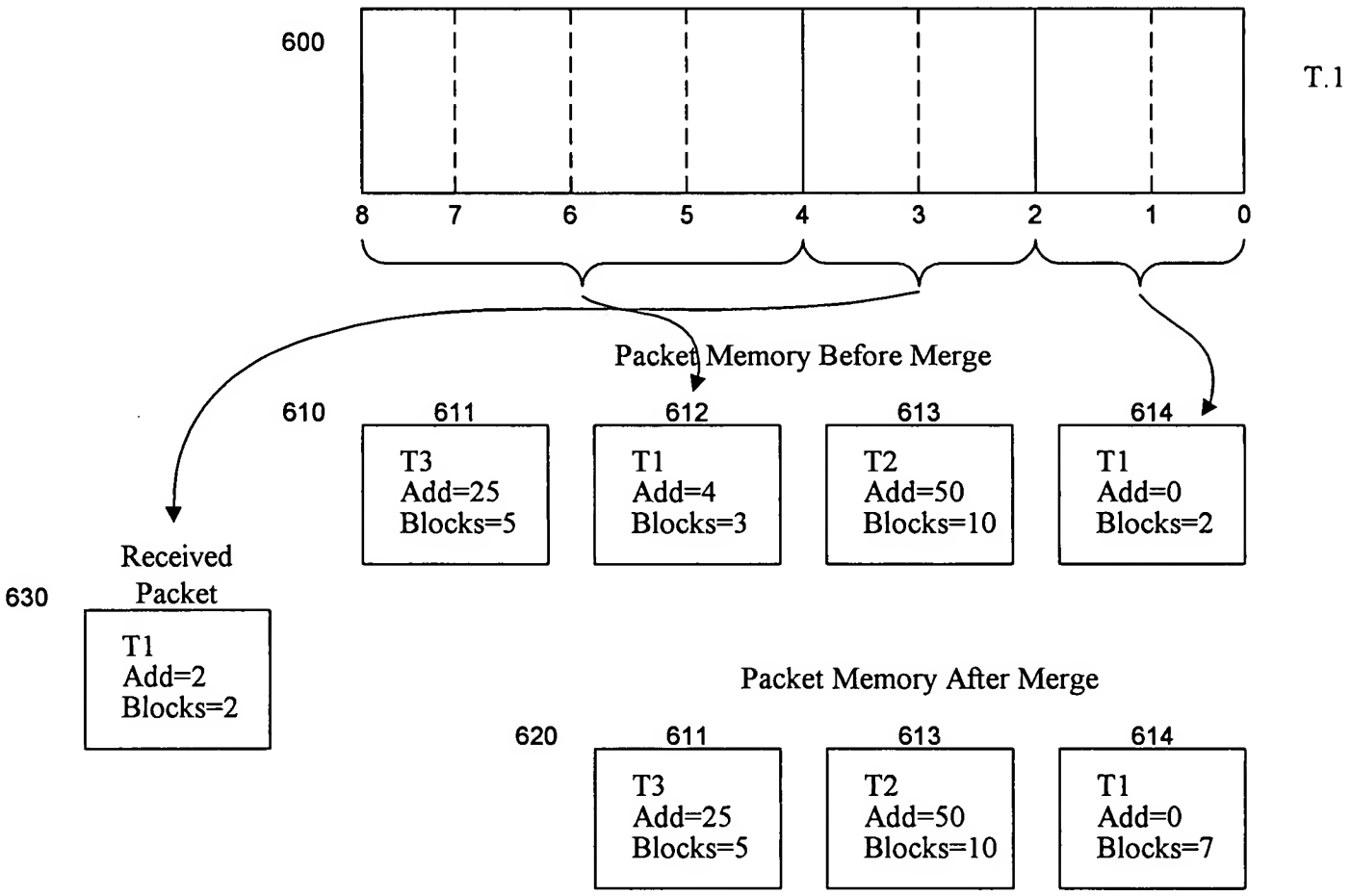



*Fig. 6*

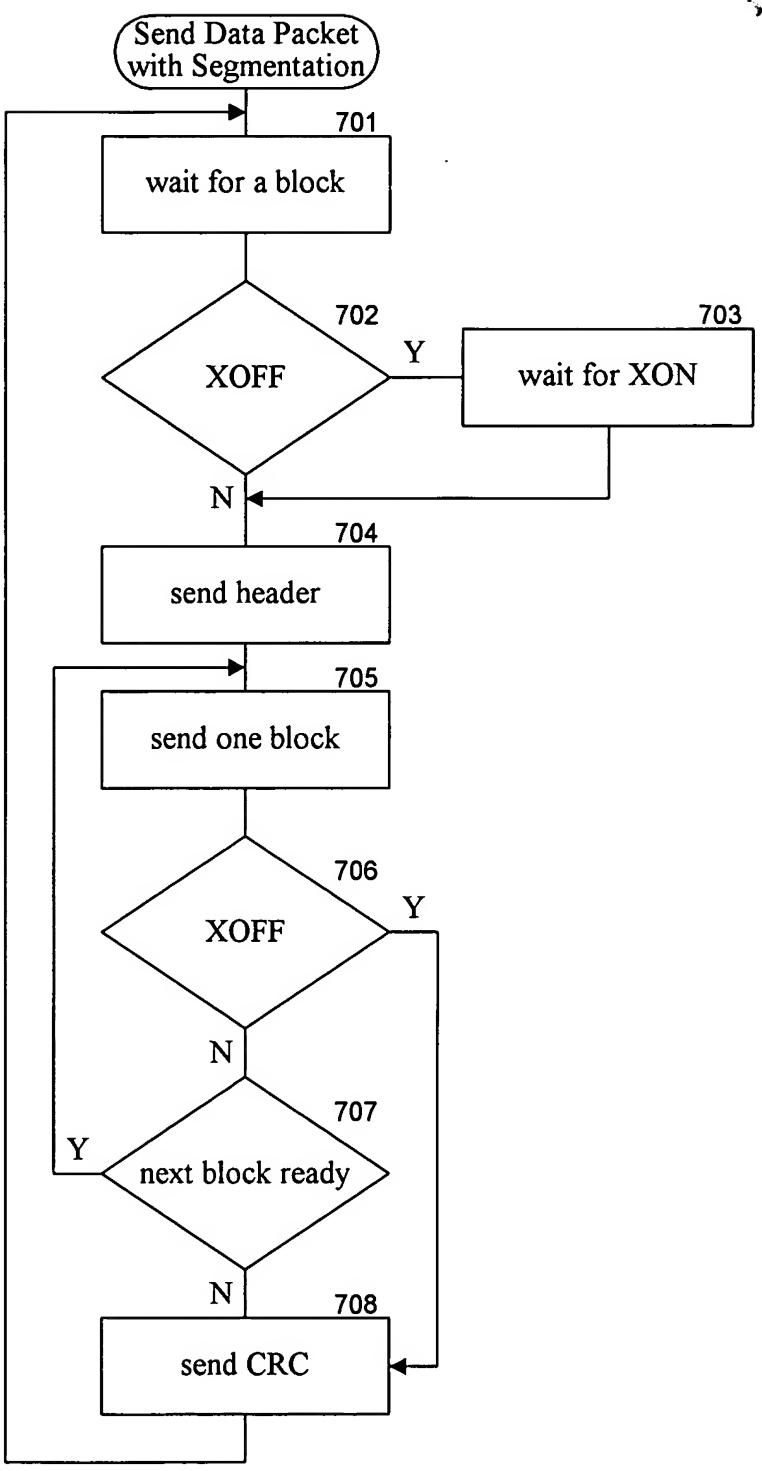

Send Data Packet with Segmentation
701
wait for a block
702
XOFF
Y
703
wait for XON
N
704
send header
705
send one block
706
XOFF
Y
N
707
Y
next block ready
N
708
send CRC

*Fig. 7*

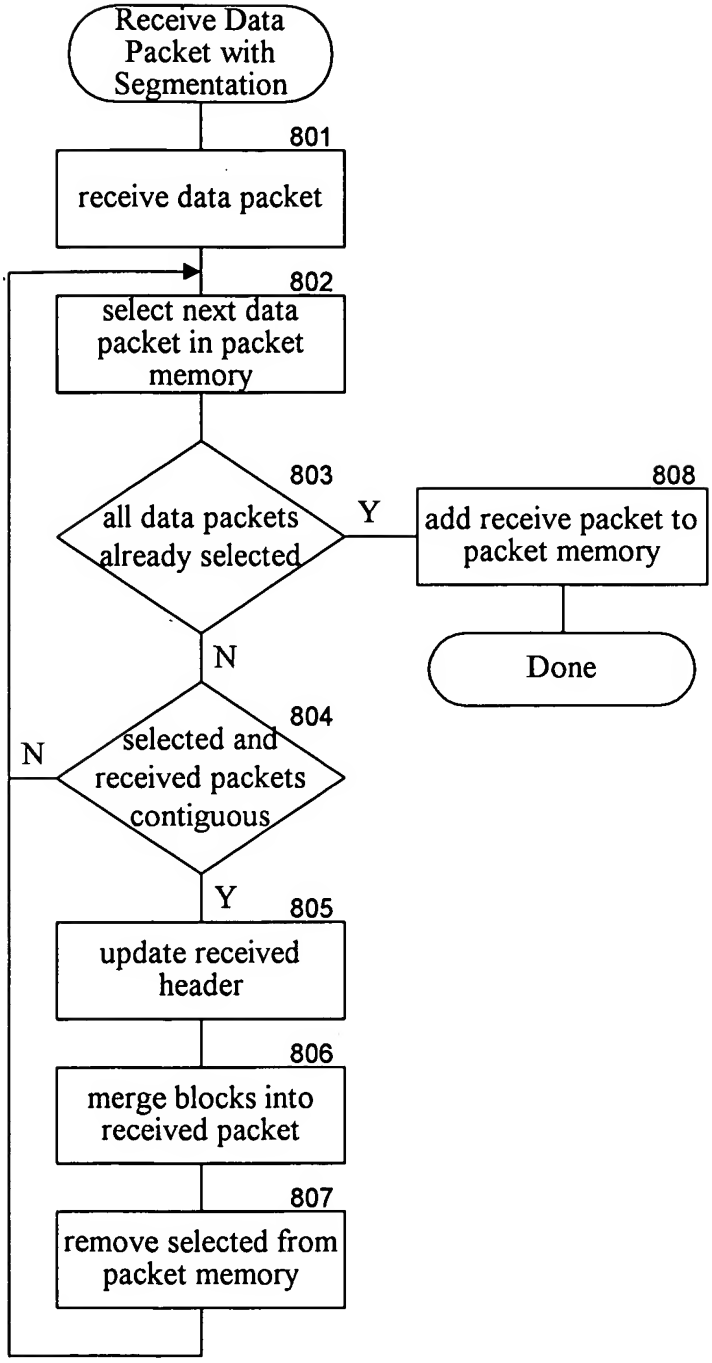

Receive Data Packet with Segmentation
801
receive data packet
802
select next data packet in packet memory
803
all data packets already selected
Y
808
add receive packet to packet memory
Done
N
804
selected and received packets contiguous
N
Y
805
update received header
806
merge blocks into received packet
807
remove selected from packet memory

*Fig. 8*

20STh0"h67ZE00T

sync & packet type

***Fig. 9A***

20STh0" h6796007

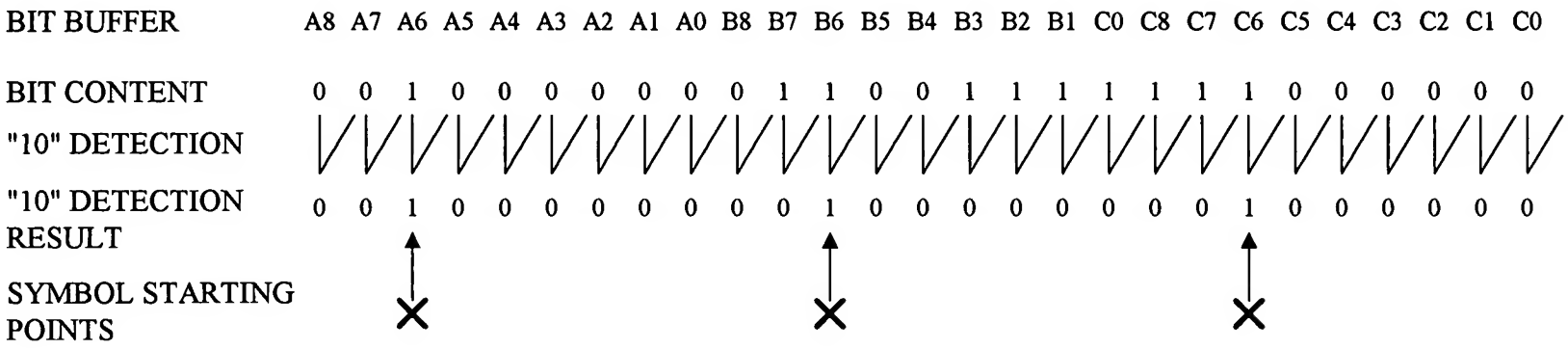



*Fig. 9B*

20STh0"h6Z9E00T

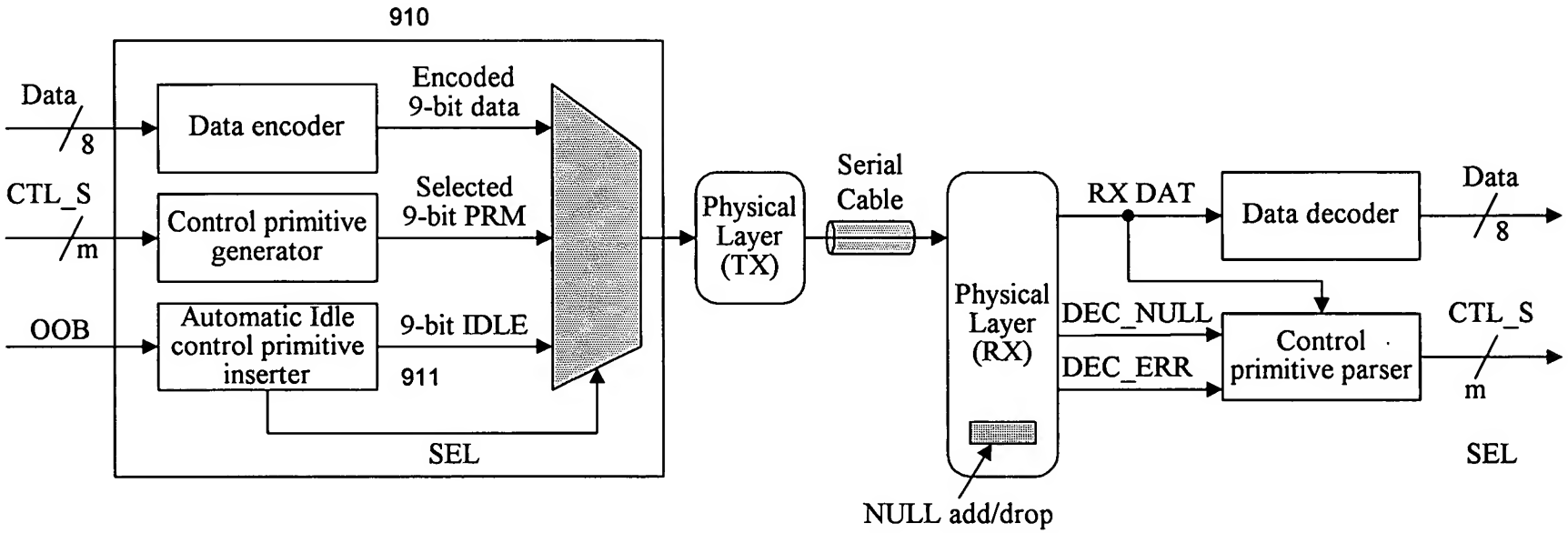



*Fig. 9C*

205Th0"h6Z9E00t

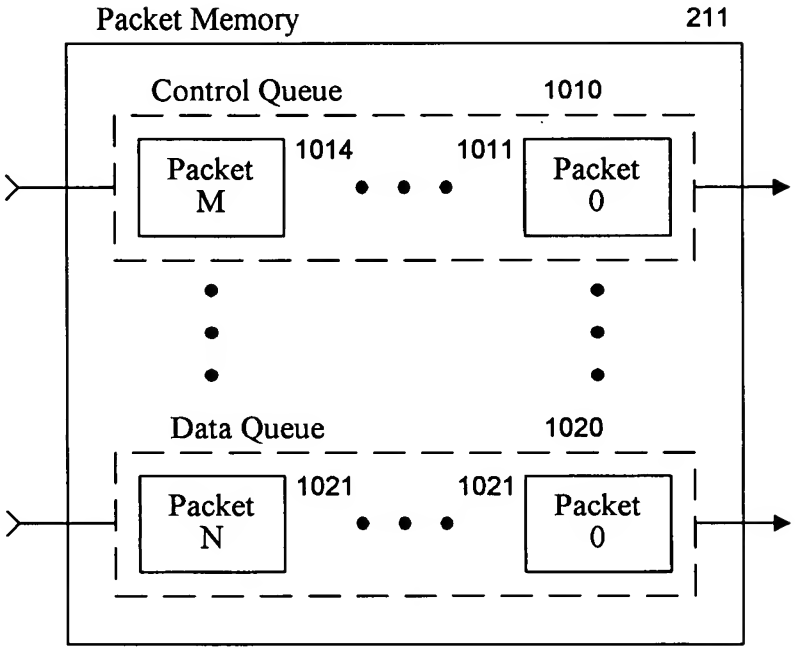



*Fig. 10*

10036704.045502

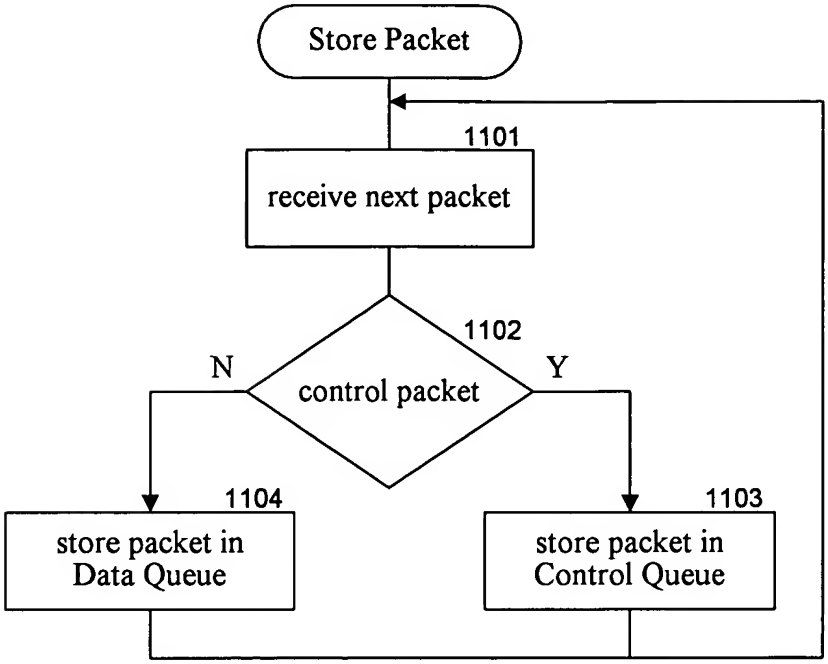



*Fig. 11*

4036704 : 04504

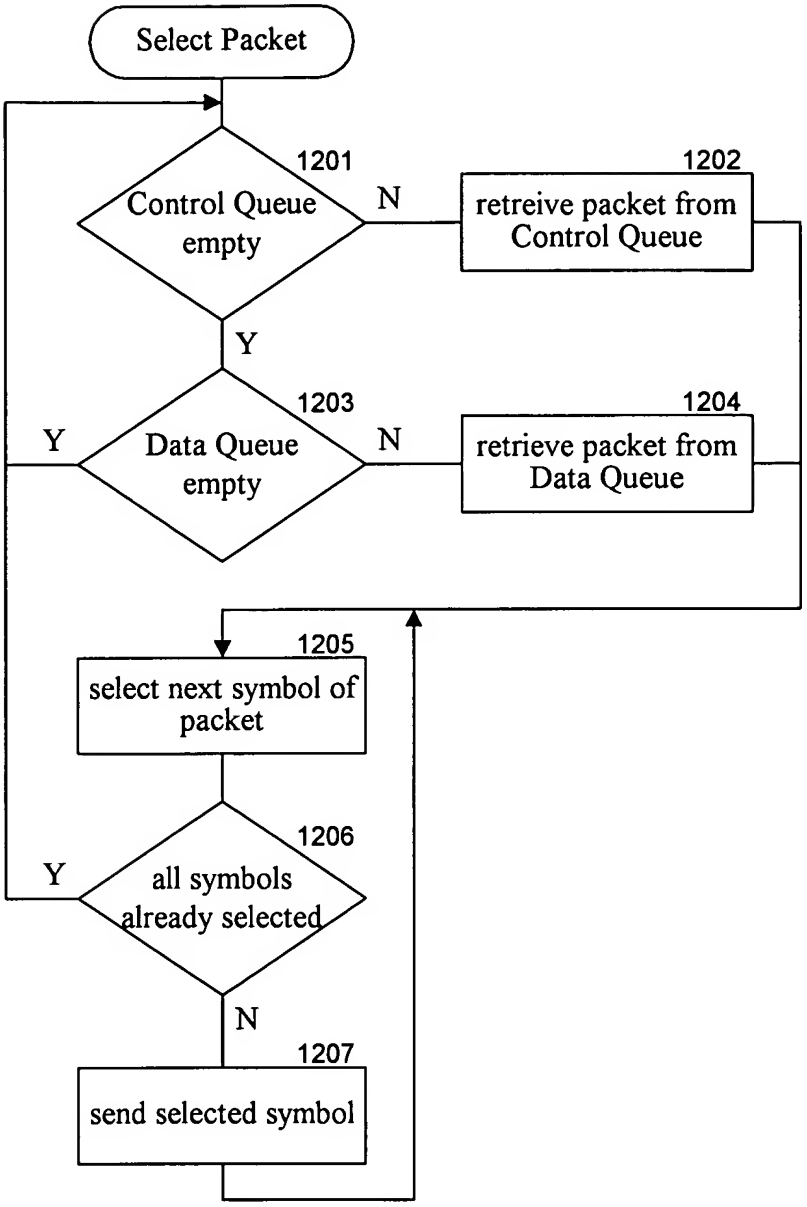



*Fig. 12*

US 6,798,739 B1

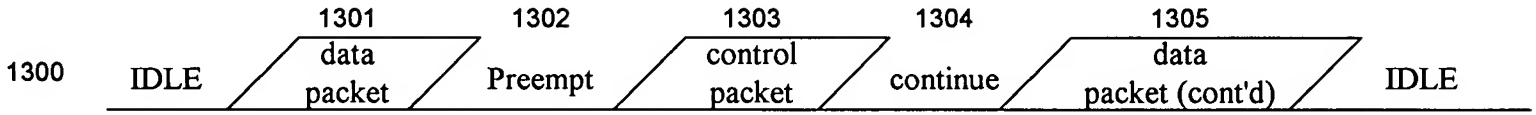



*Fig. 13*

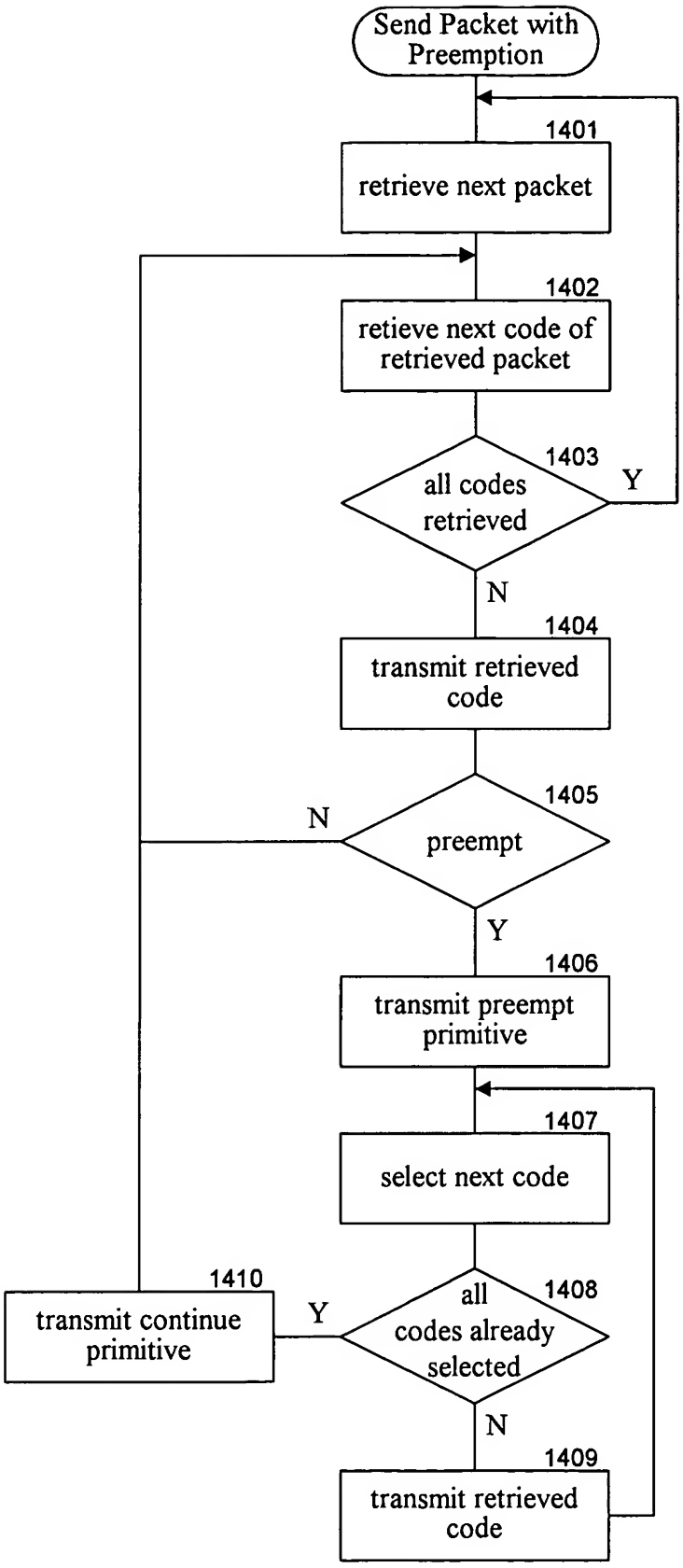

Send Packet with Preemption
1401
retrieve next packet
1402
retieve next code of retrieved packet
1403
all codes retrieved
Y
N
1404
transmit retrieved code
1405
preempt
N
Y
1406
transmit preempt primitive
1407
select next code
1408
all codes already selected
Y
1410
transmit continue primitive
N
1409
transmit retrieved code

*Fig. 14*

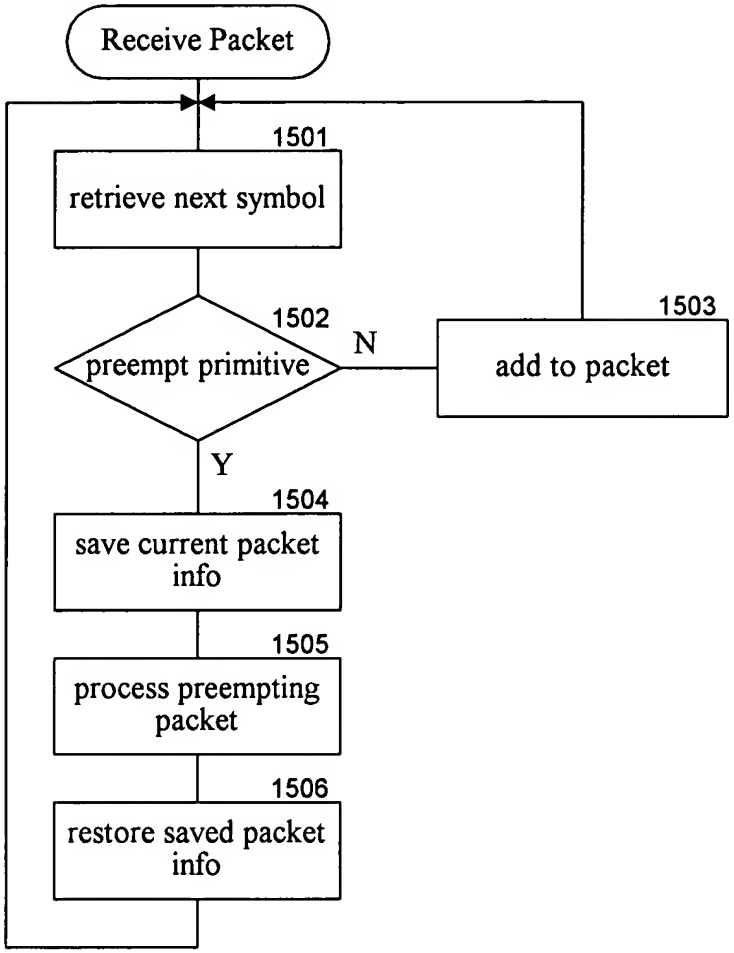

Receive Packet
1501
retrieve next symbol
1502
preempt primitive
N
1503
add to packet
Y
1504
save current packet info
1505
process preempting packet
1506
restore saved packet info

*Fig. 15*

20STh0" h6Z9E00T

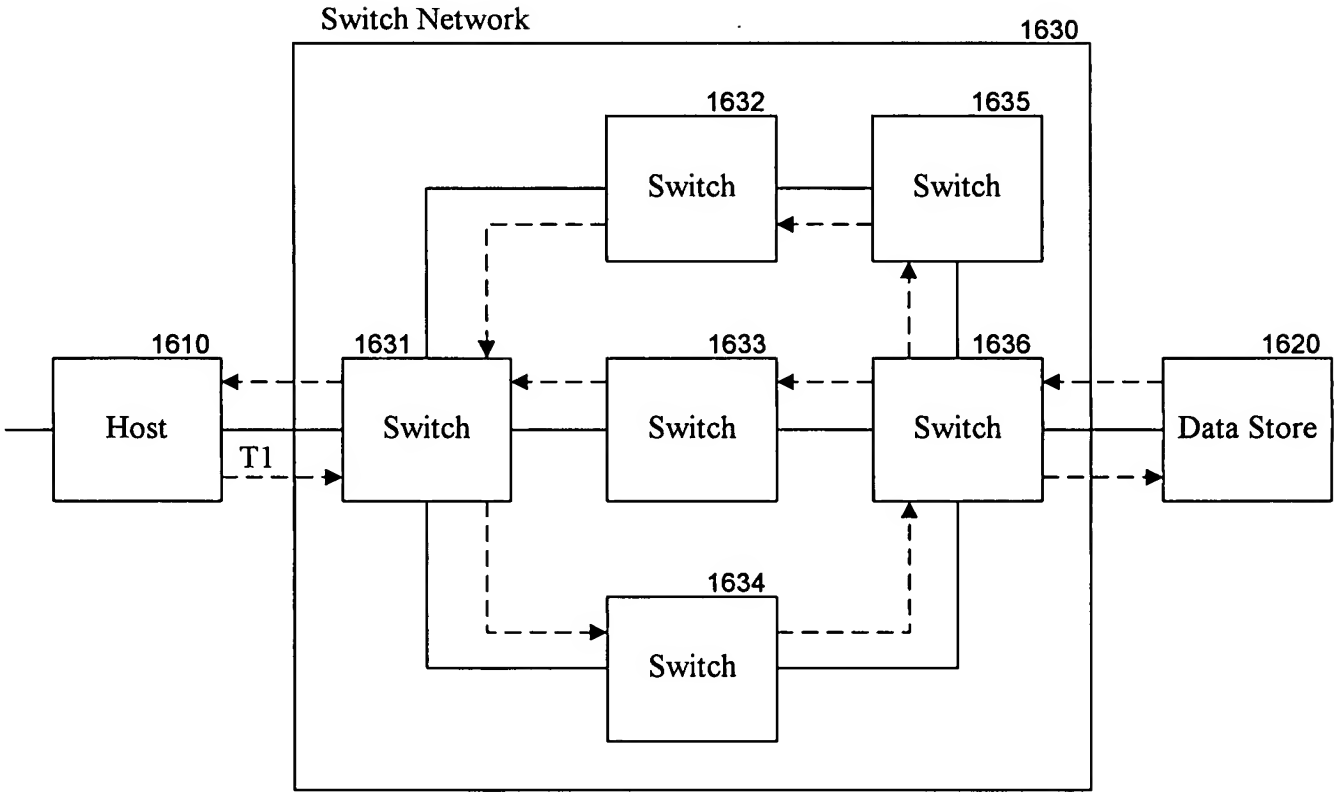



*Fig. 16*

20STh0"h6Z9E00T

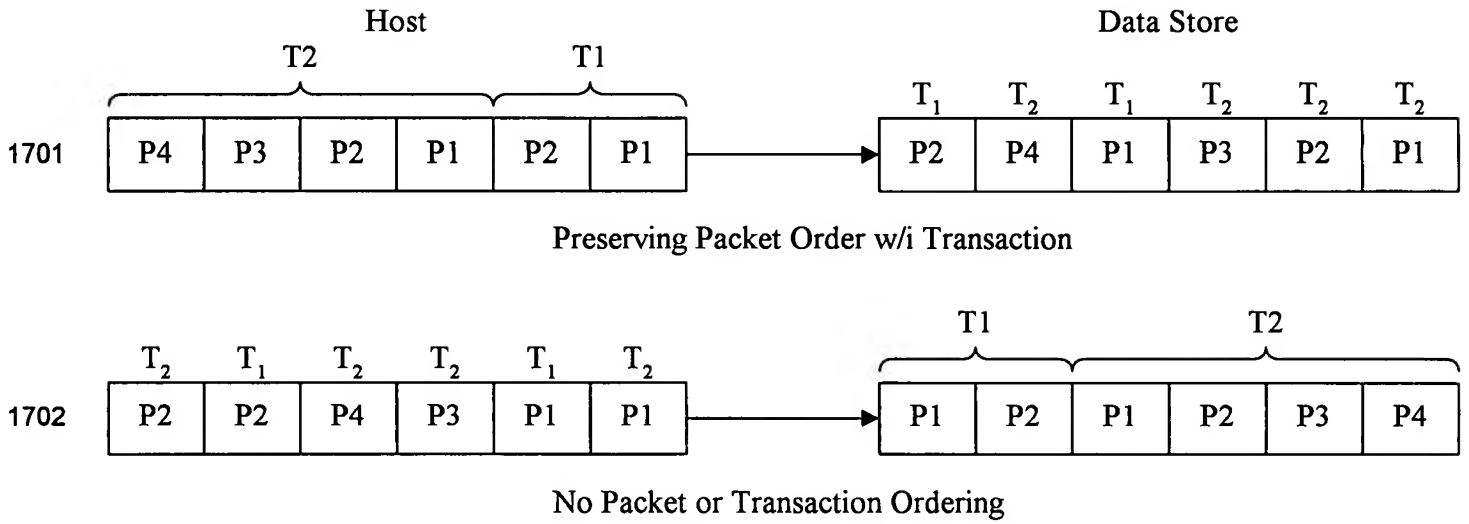



*Fig. 17*

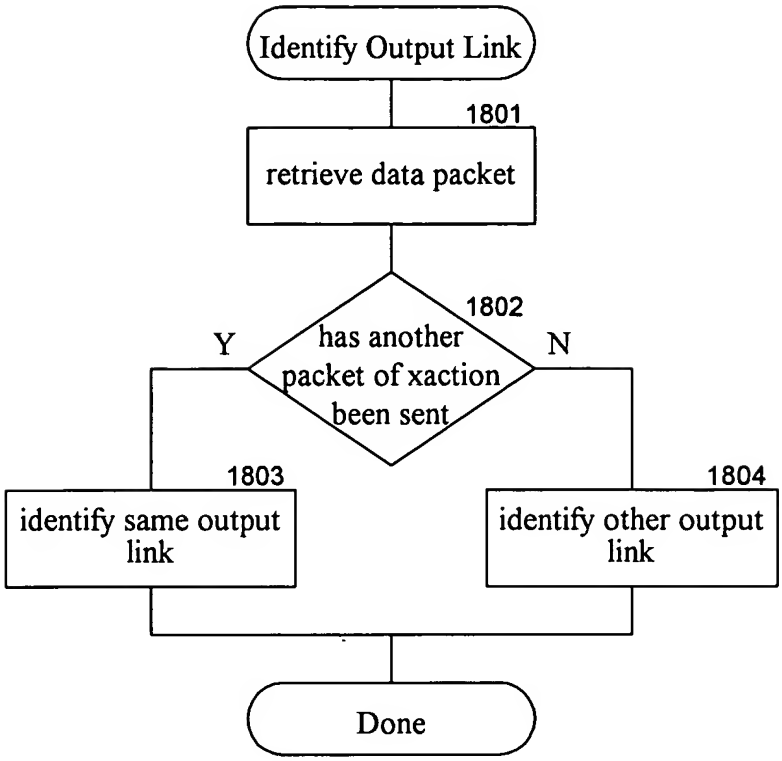

Identify Output Link
1801
retrieve data packet
1802
has another packet of xaction been sent
Y
N
1803
identify same output link
1804
identify other output link
Done

*Fig. 18*

205th0"h6z9E00f

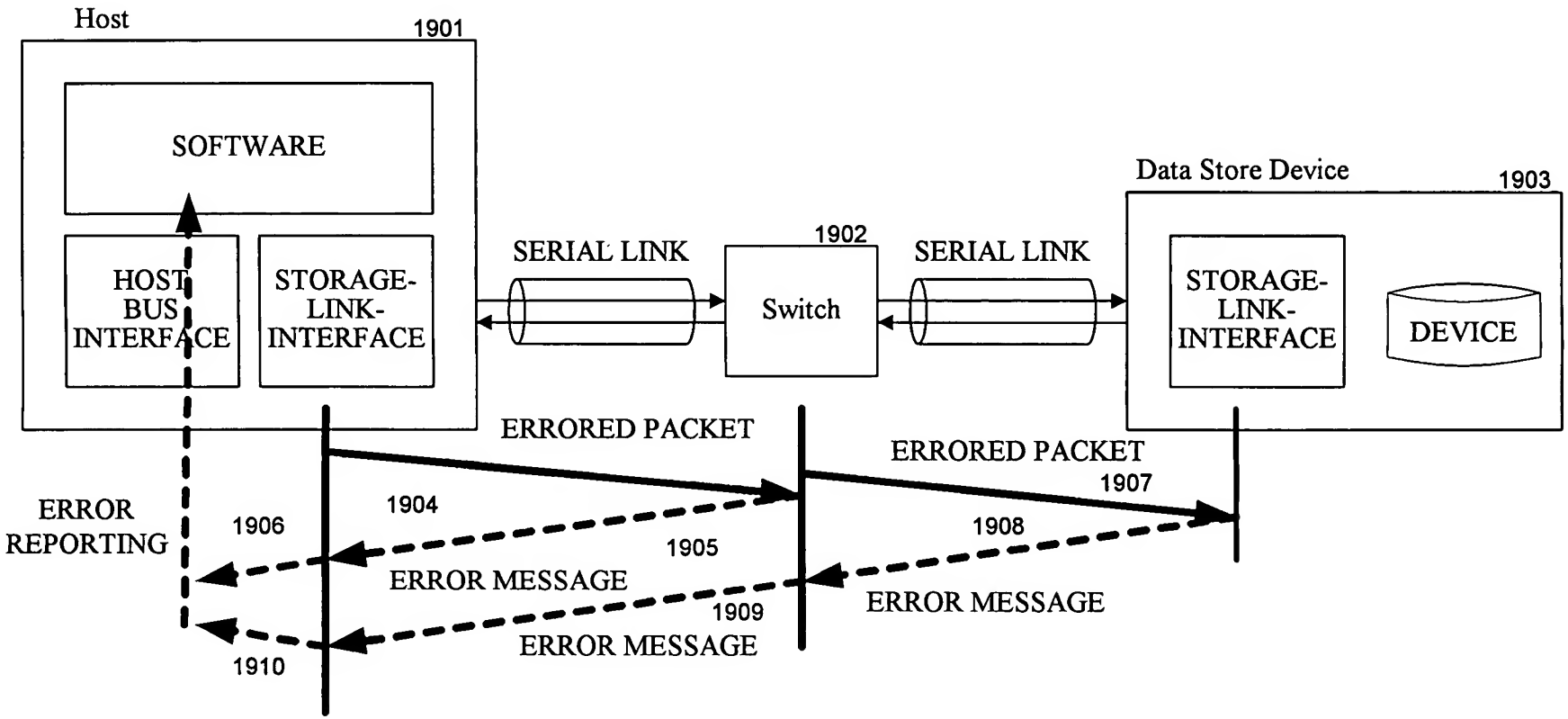



*Fig. 19A*

205Th0" h67960Of

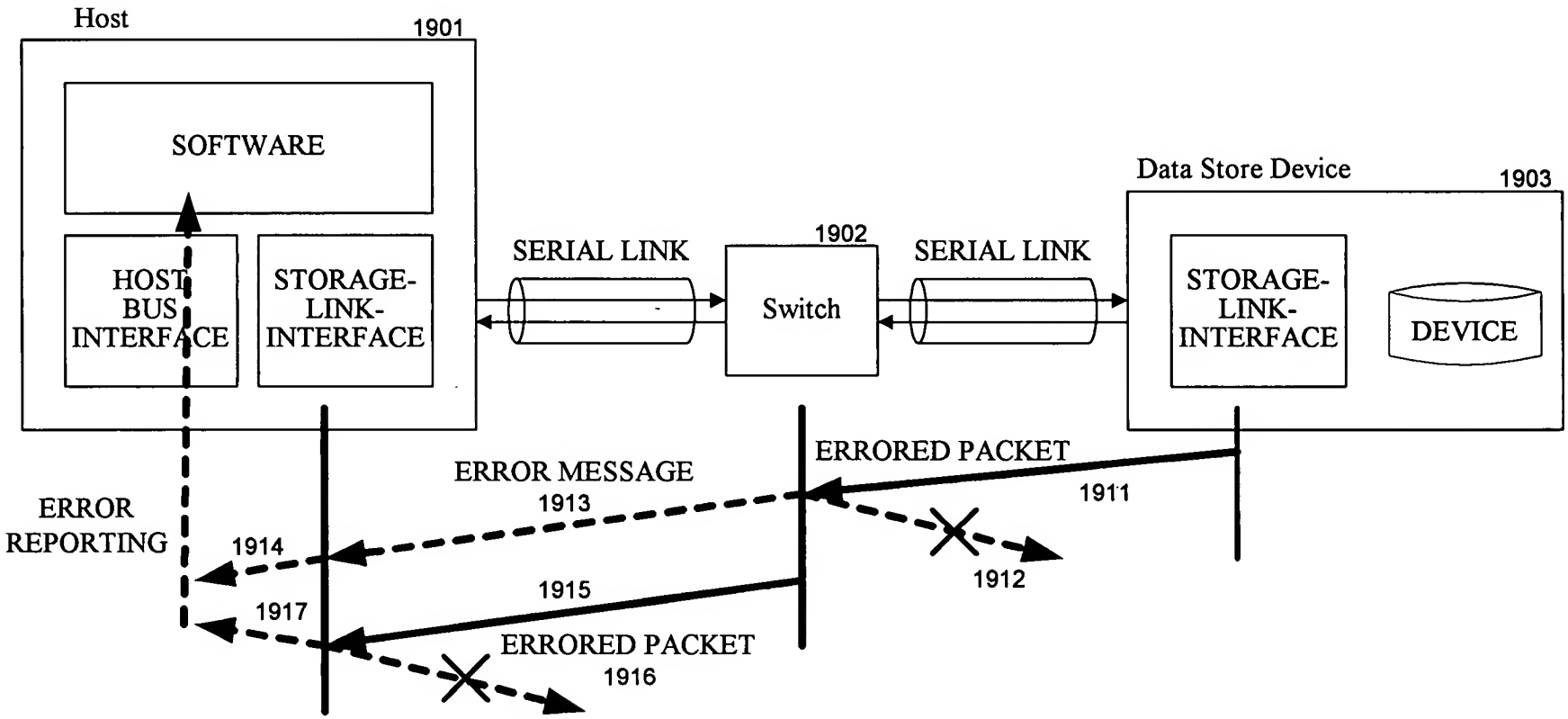



*Fig. 19B*

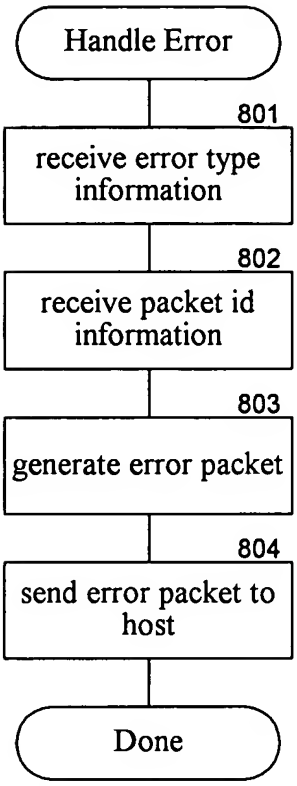

Handle Error
801
receive error type information
802
receive packet id information
803
generate error packet
804
send error packet to host
Done

*Fig. 19C*

| 8b code | 9 bit symbol |
|---|---|
| 0000 0000 | 101010101 |
| 0000 0001 | 101010100 |
| 0000 0010 | 101010111 |
| ⋮ | |
| 0101 0101 | 001010101 |
| ⋮ | |
| 0111 0110 | 001110110 |
| 0111 0111 | 100100010 |
| ⋮ | |
| 1111 1111 | 110101010 |

*Fig. 20*

2004054950 - 04-03-2004

205Th0"h6Z9E00F

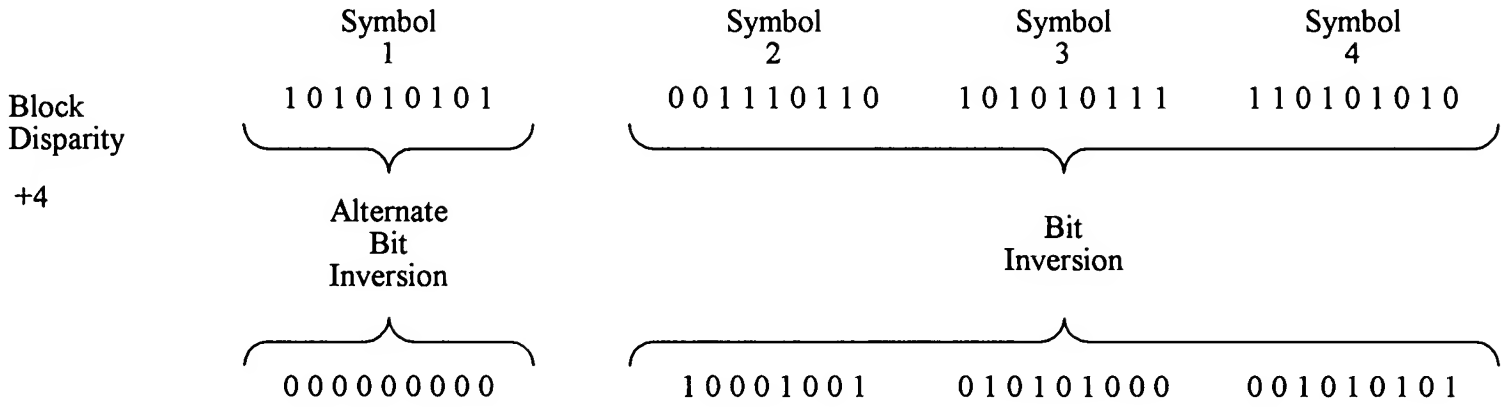



*Fig. 21A*

20STh0" h5Z9E00T

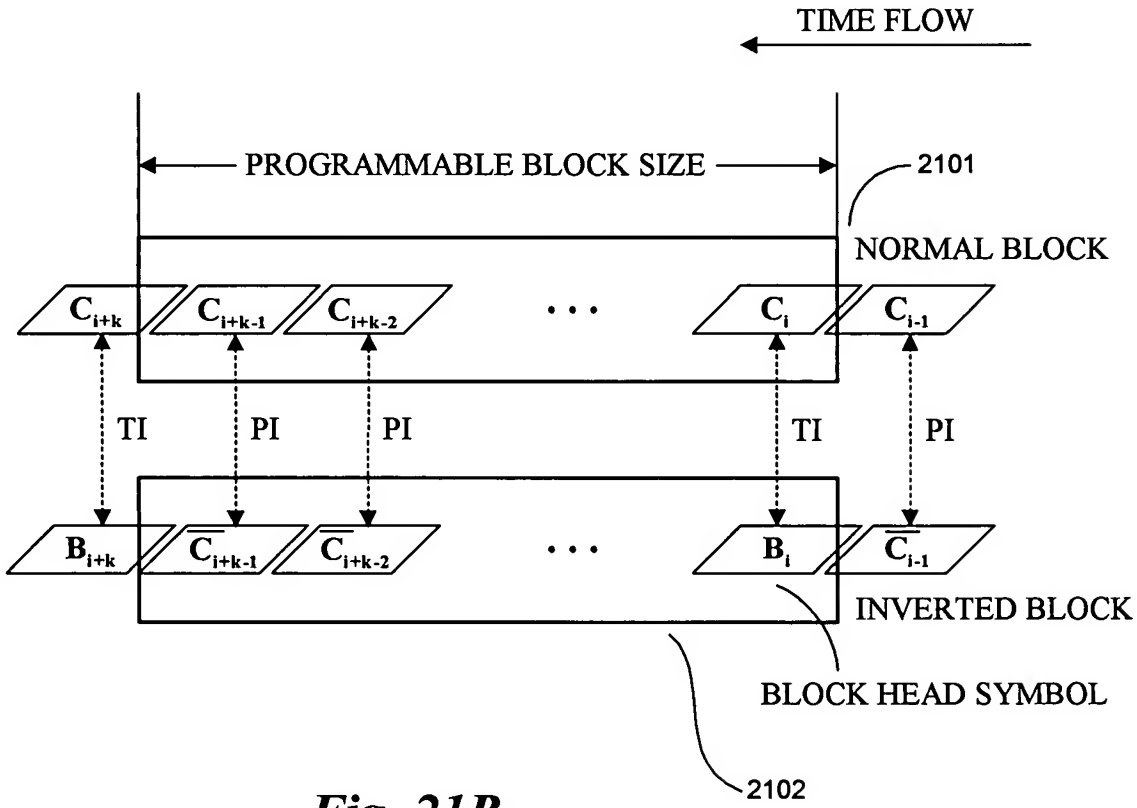



*Fig. 21B*

20STh0" Th6Z9E00T

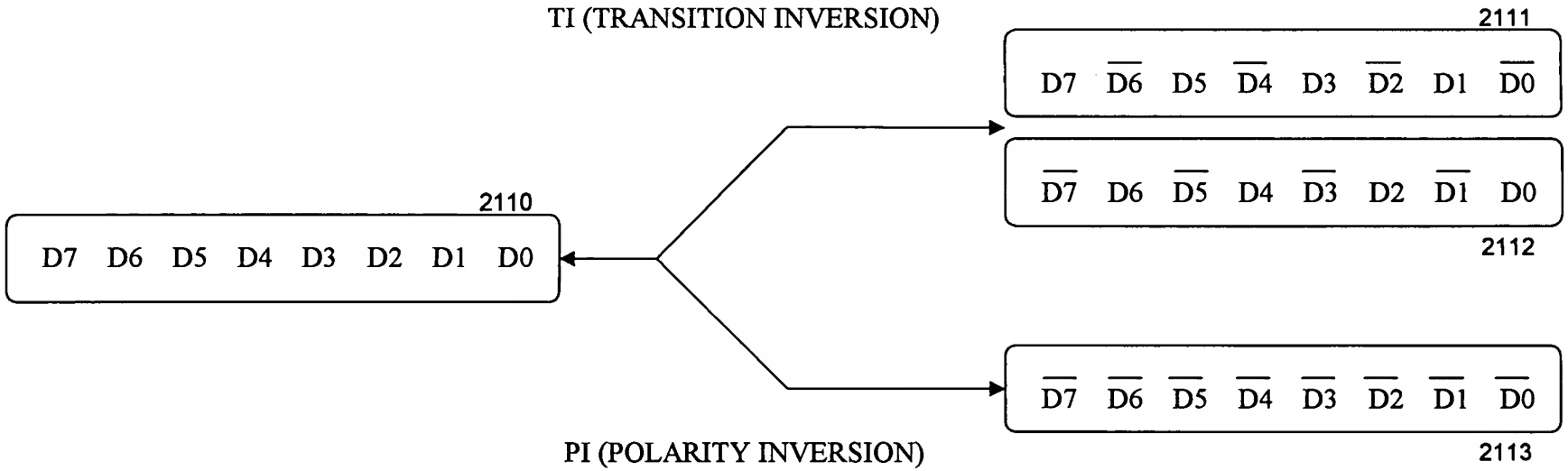



*Fig. 21C*

20STh0" h6Z9E0QT

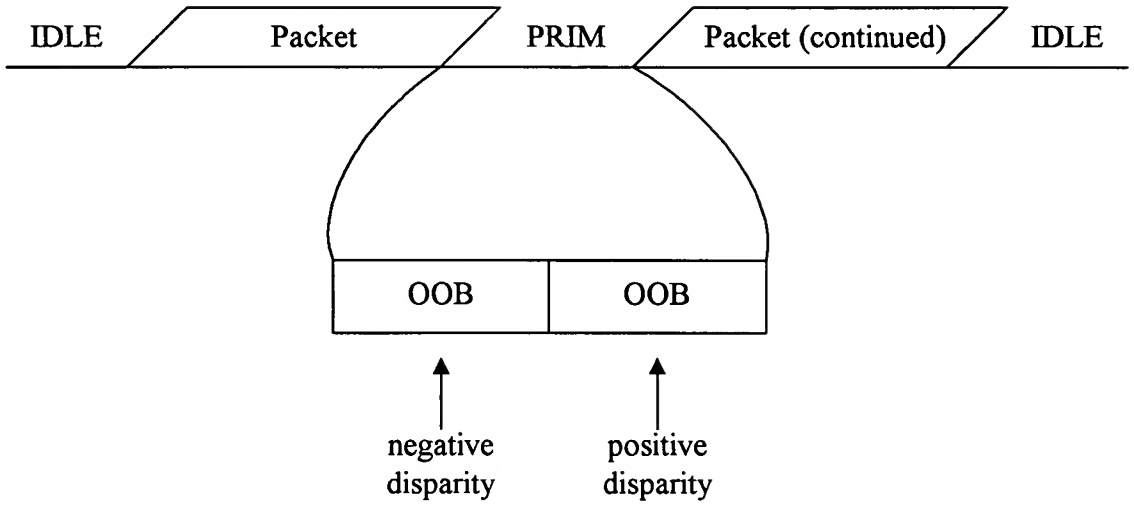



*Fig. 22*

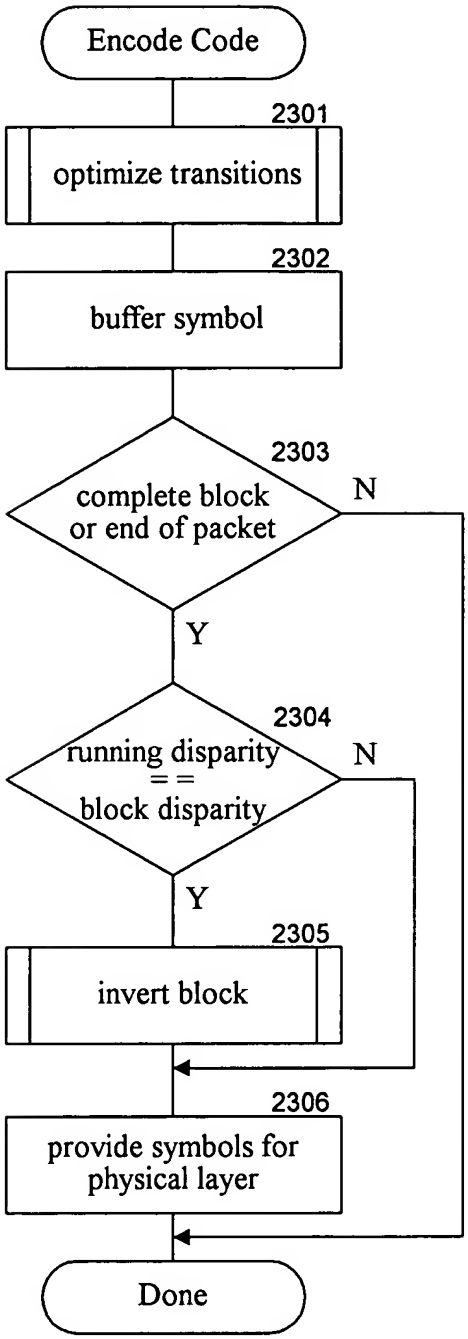

Encode Code
2301
optimize transitions
2302
buffer symbol
2303
complete block or end of packet
N
Y
2304
running disparity == block disparity
N
Y
2305
invert block
2306
provide symbols for physical layer
Done

*Fig. 23*

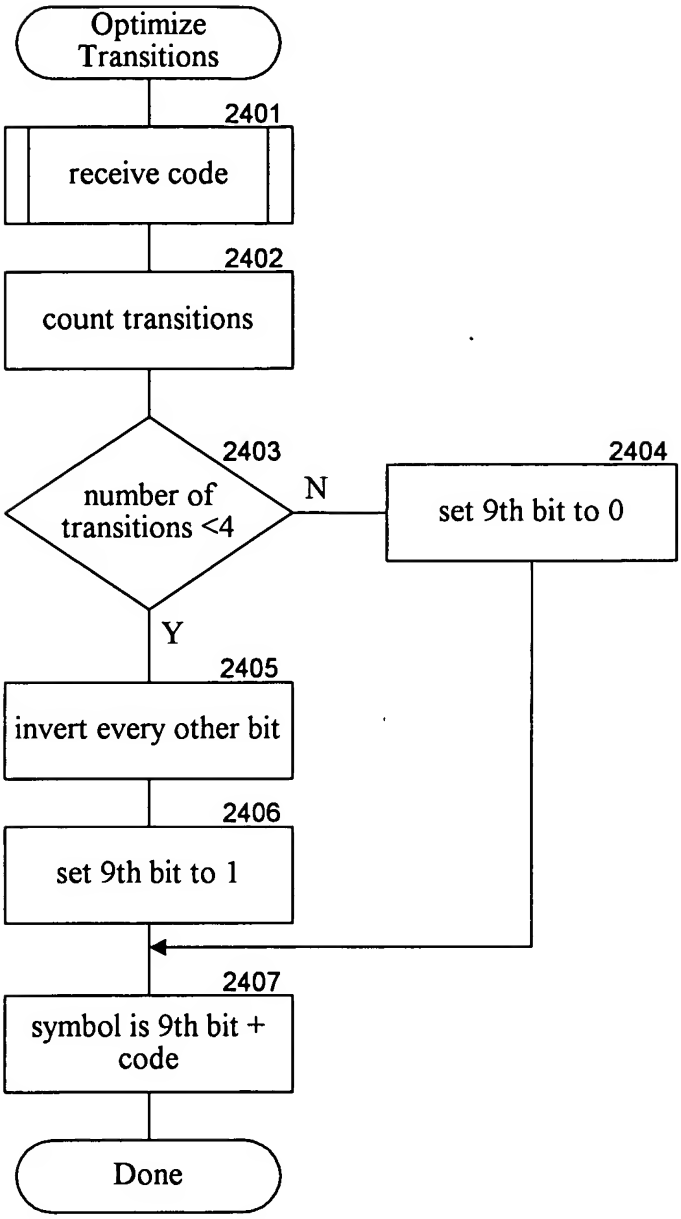

Optimize Transitions
2401
receive code
2402
count transitions
2403
number of transitions <4
N
2404
set 9th bit to 0
Y
2405
invert every other bit
2406
set 9th bit to 1
2407
symbol is 9th bit + code
Done

*Fig. 24*

EXPRESS MAIL NO. EV 339076405 US

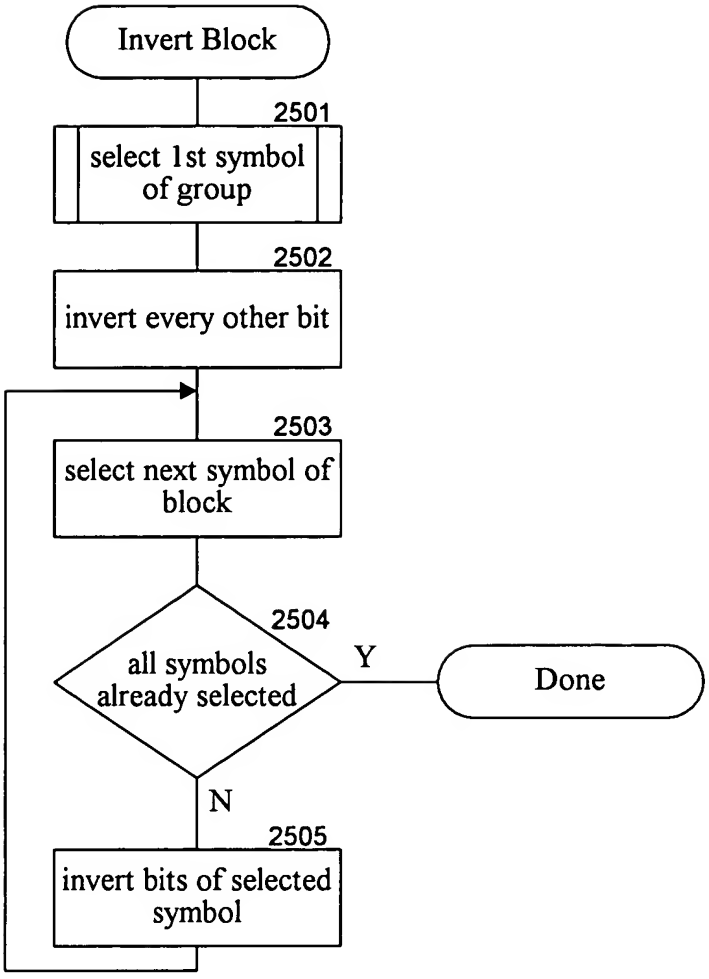



*Fig. 25*

20050124.0300

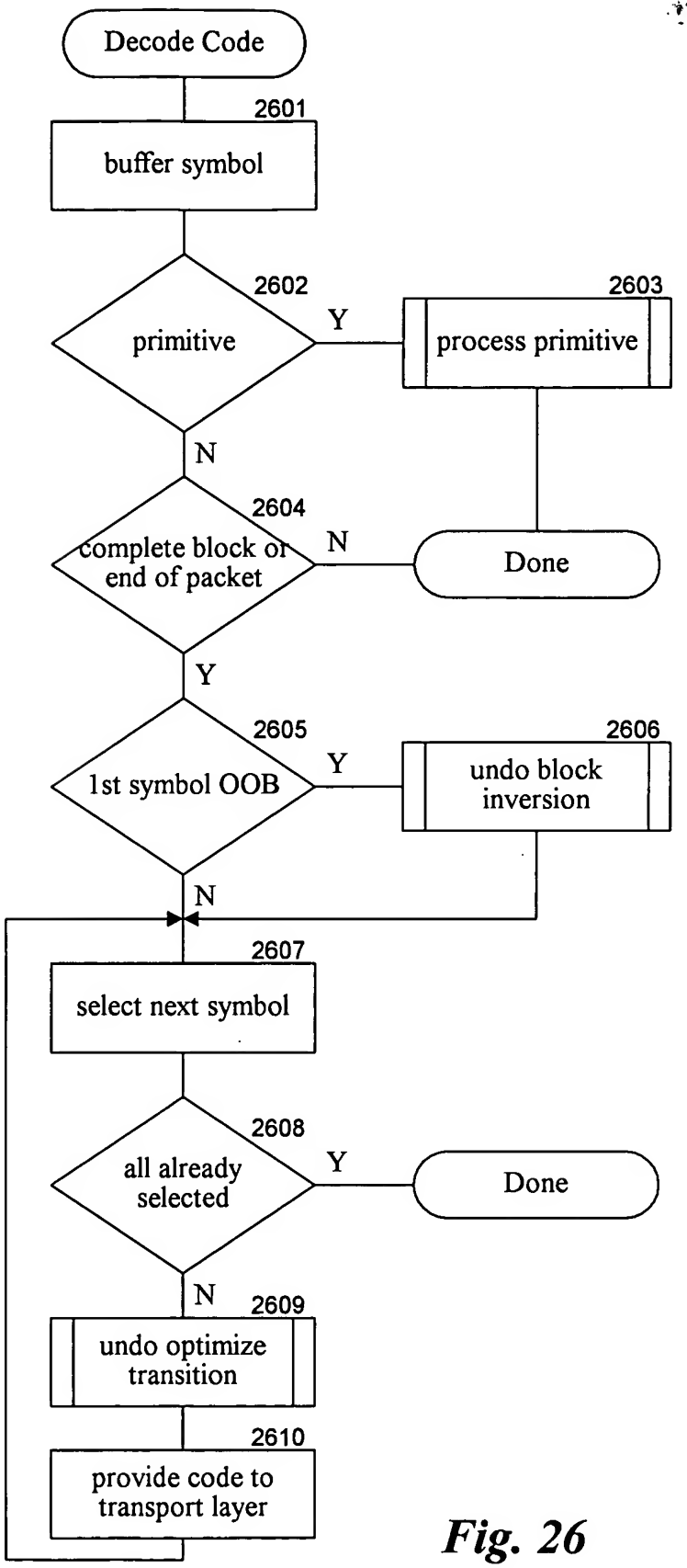



*Fig. 26*

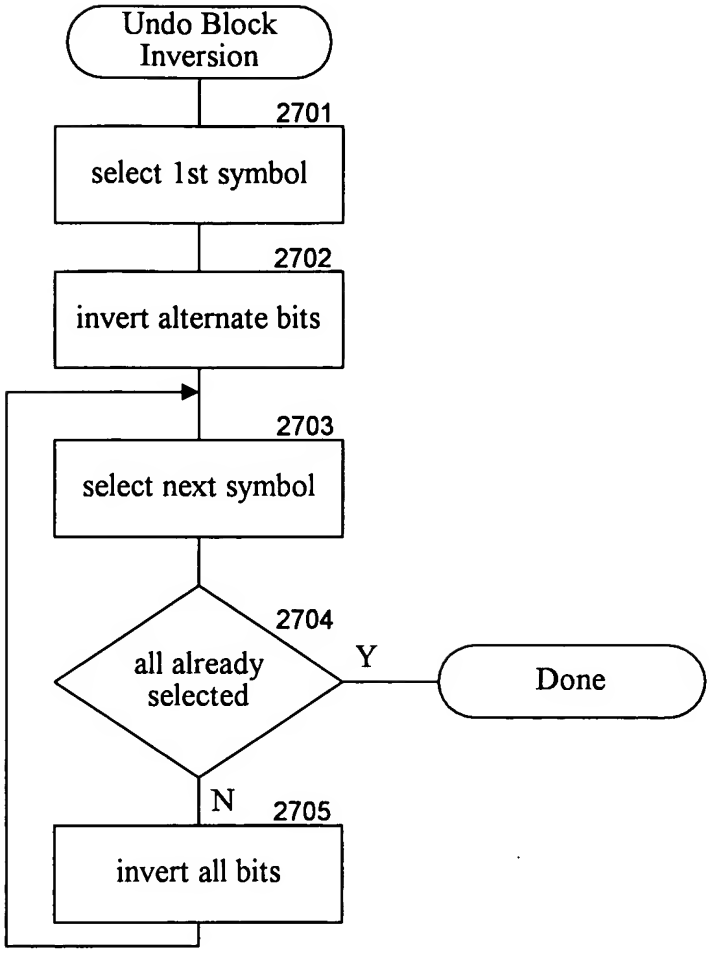

Undo Block Inversion
2701
select 1st symbol
2702
invert alternate bits
2703
select next symbol
2704
all already selected
Y
Done
N
2705
invert all bits

*Fig. 27*

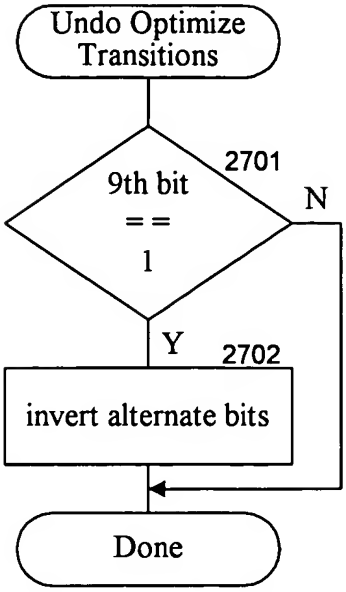

Undo Optimize Transitions
9th bit == 1
2701
N
Y
2702
invert alternate bits
Done

*Fig. 28*

4005345.0402

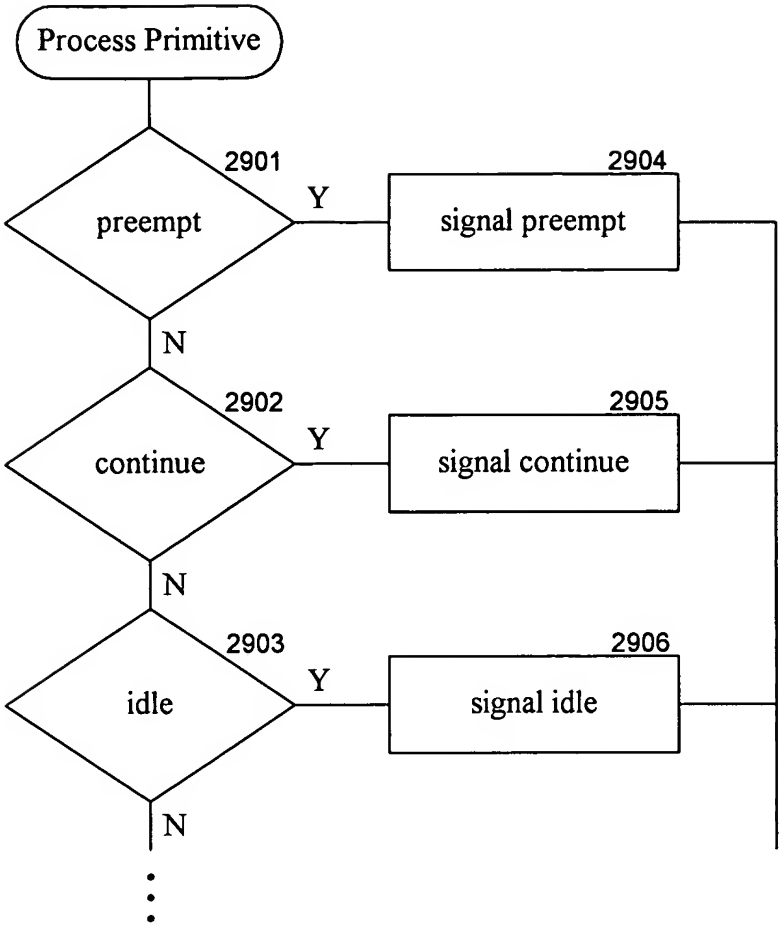



*Fig. 29*

20STh0"h6.Z9E00T

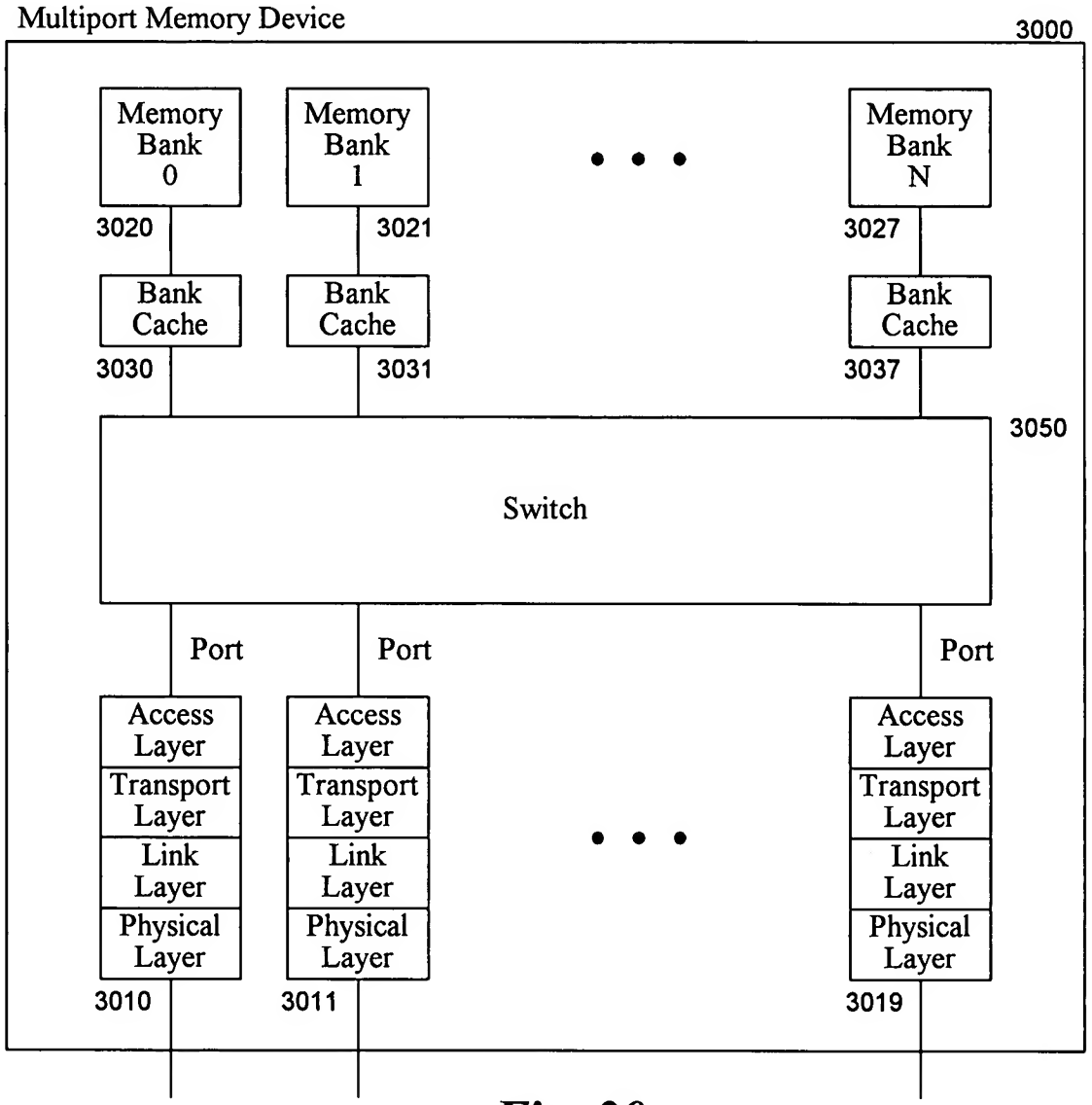



*Fig. 30*

20STh0"h6Z9E00T

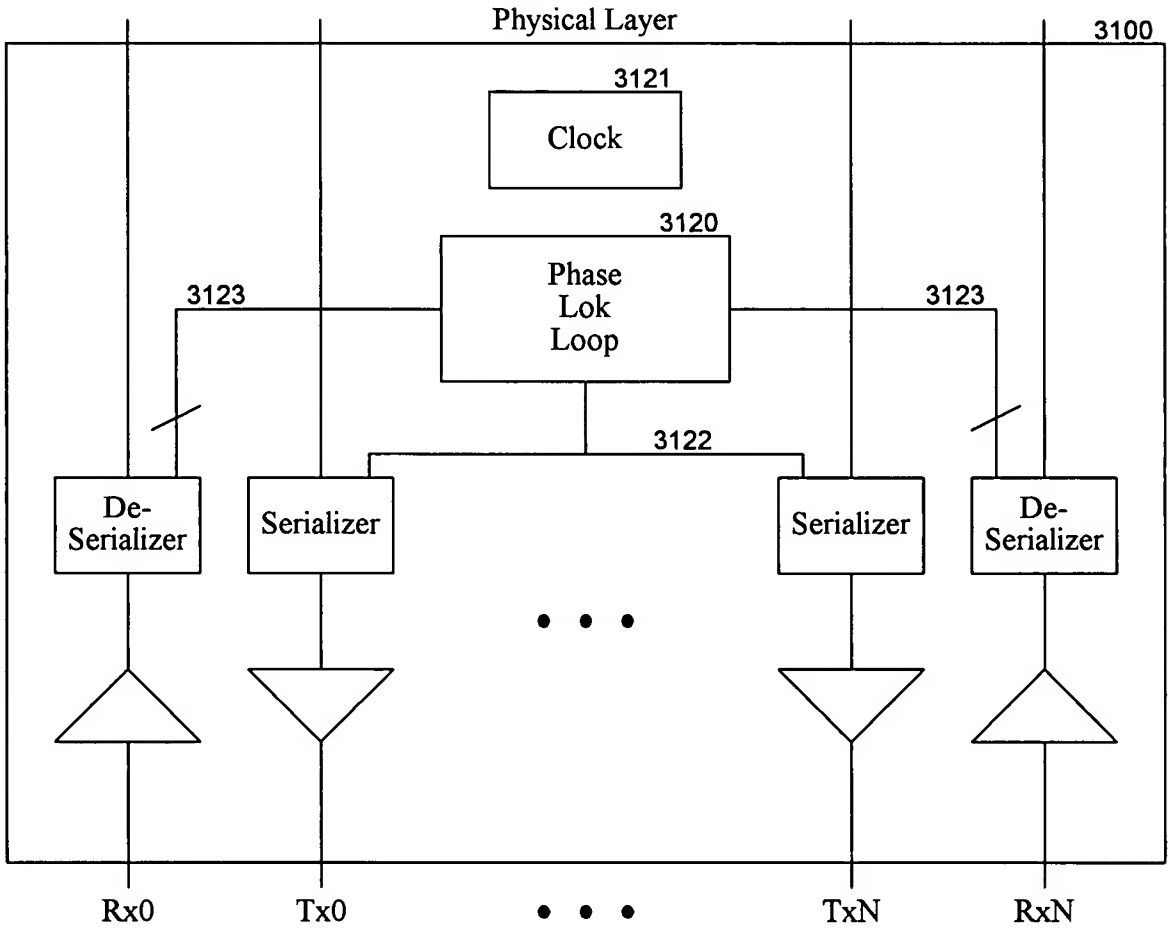



*Fig. 31*

205th0" th6z9E00t

Input Queue 3201

| Port | R/W | Address | Data |
|---|---|---|---|
| 3 | R | 1000 | |
| 4 | W | 4000 | 10....1 |
| 3 | W | 1000 | 111...0 |
| 3 | R | 2000 | |
| | ⋮ | | |

Output Queue 3202

| Valid | Port | Data |
|---|---|---|
| 1 | 3 | 11...0 |
| 0 | | |
| 0 | | |
| 1 | 3 | 101...1 |
| | ⋮ | |

*Fig. 32*

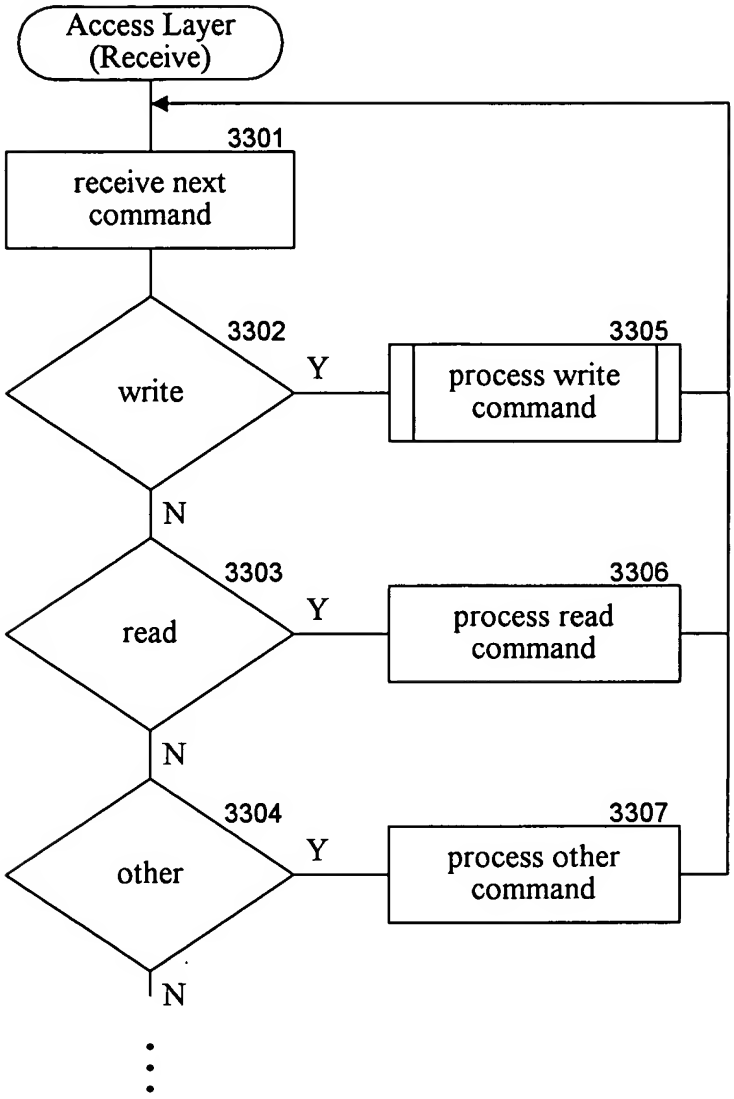

Access Layer
(Receive)
3301
receive next
command
3302
write
Y
3305
process write
command
N
3303
read
Y
3306
process read
command
N
3304
other
Y
3307
process other
command
N

*Fig. 33*

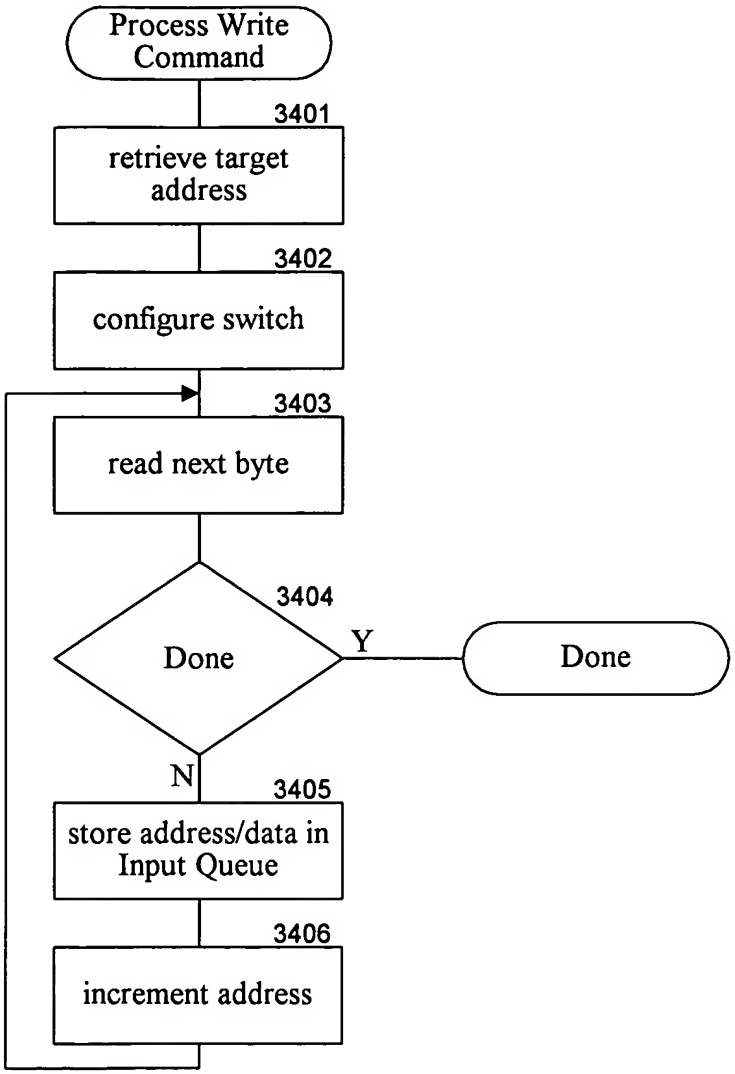

Process Write Command
3401
retrieve target address
3402
configure switch
3403
read next byte
3404
Done
Y
Done
N
3405
store address/data in Input Queue
3406
increment address

*Fig. 34*

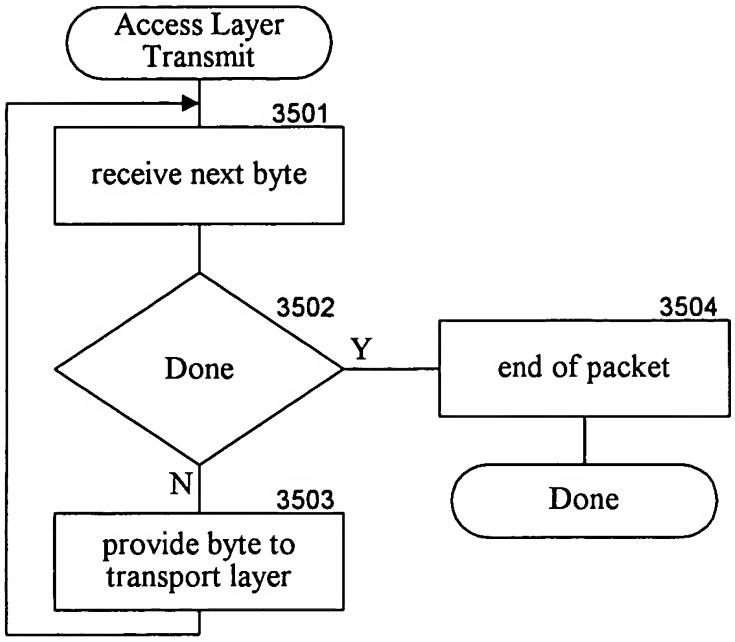



*Fig. 35*

10035504-045502

20STh0" h6298007

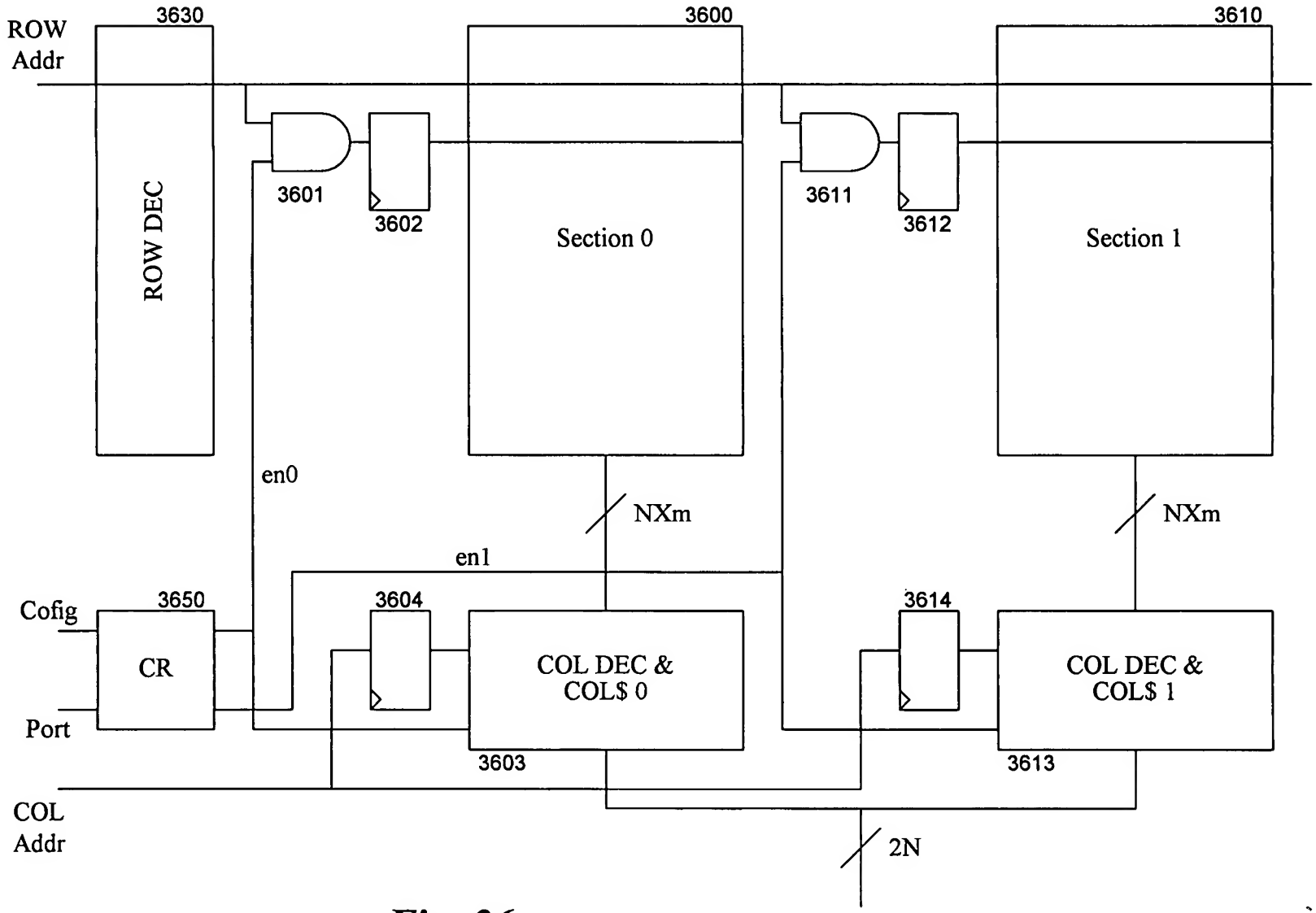
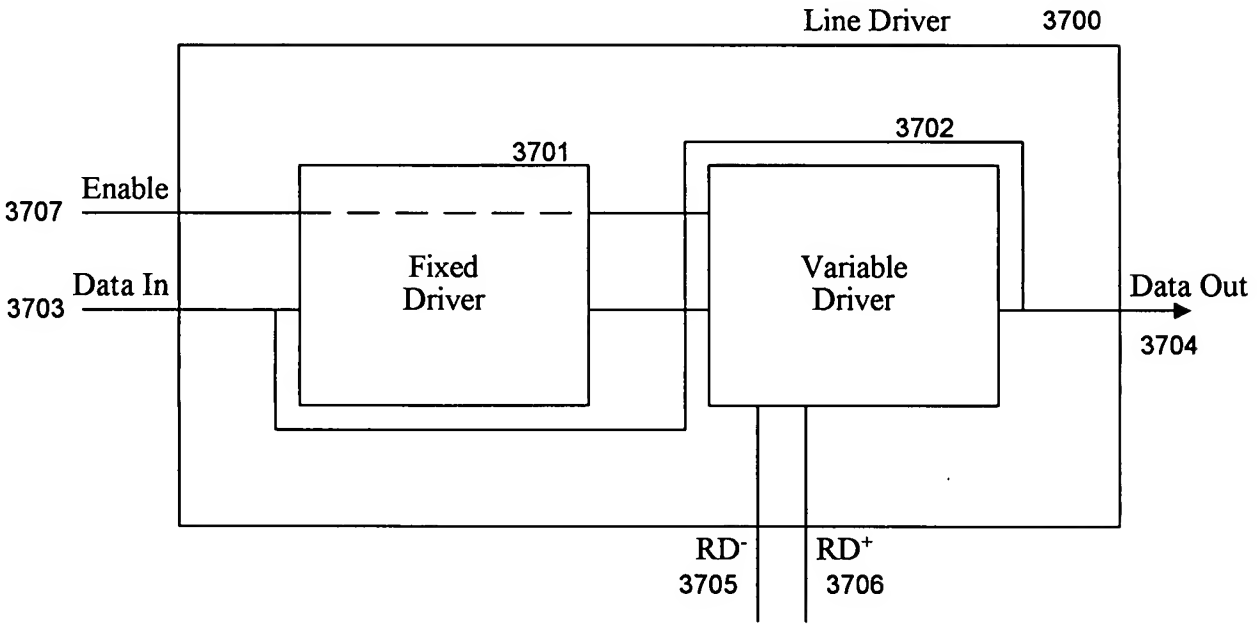



*Fig. 36*

20STh0" h629E00T




Variable Driver
- $RD^+ \wedge \overline{DataIn}$ = pull down
- $RD^- \wedge DataIn$ = pull up

*Fig. 37A*

20STh0* h629E00T

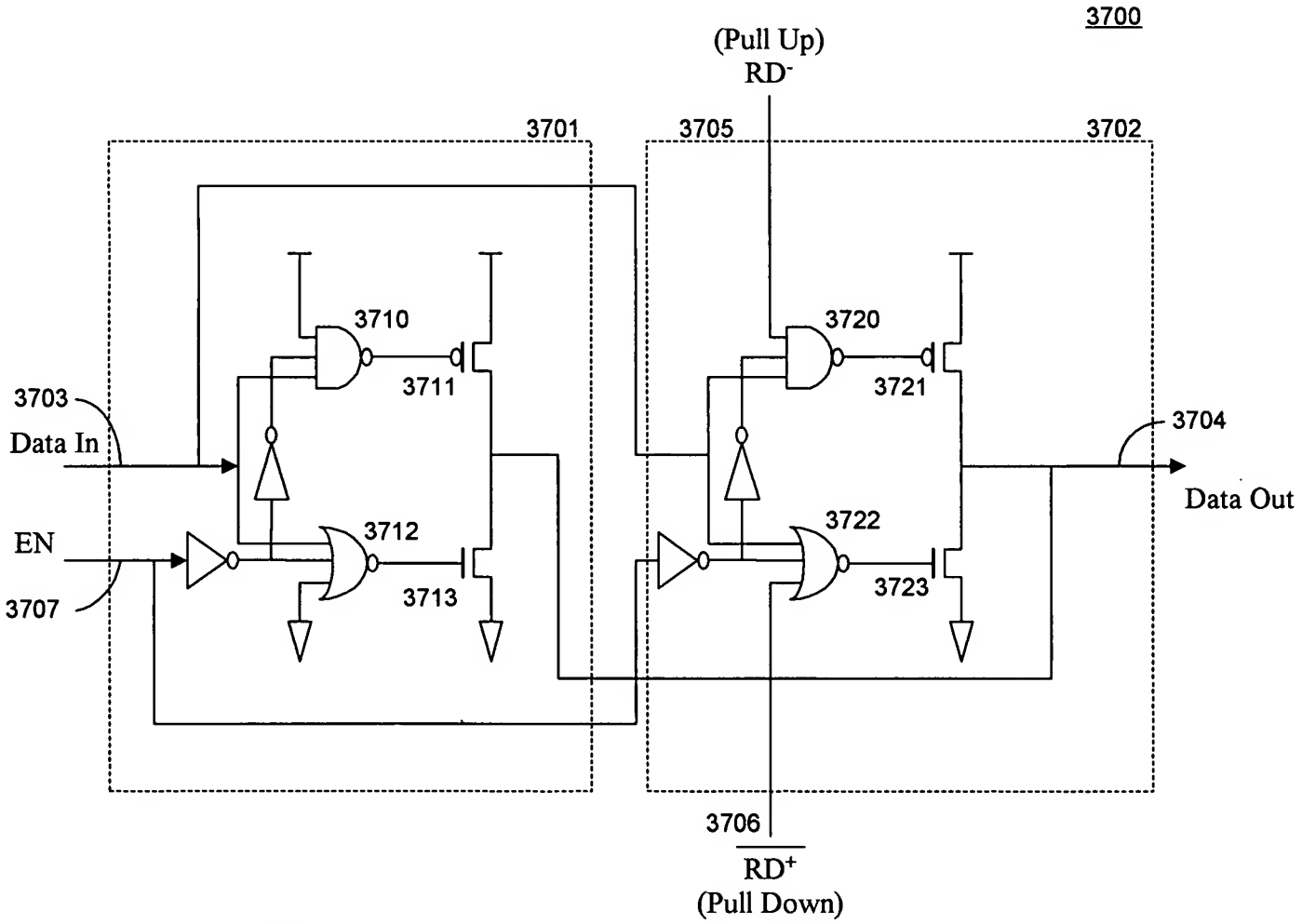



*Fig. 37B*

20STh0" th6Z9E00T

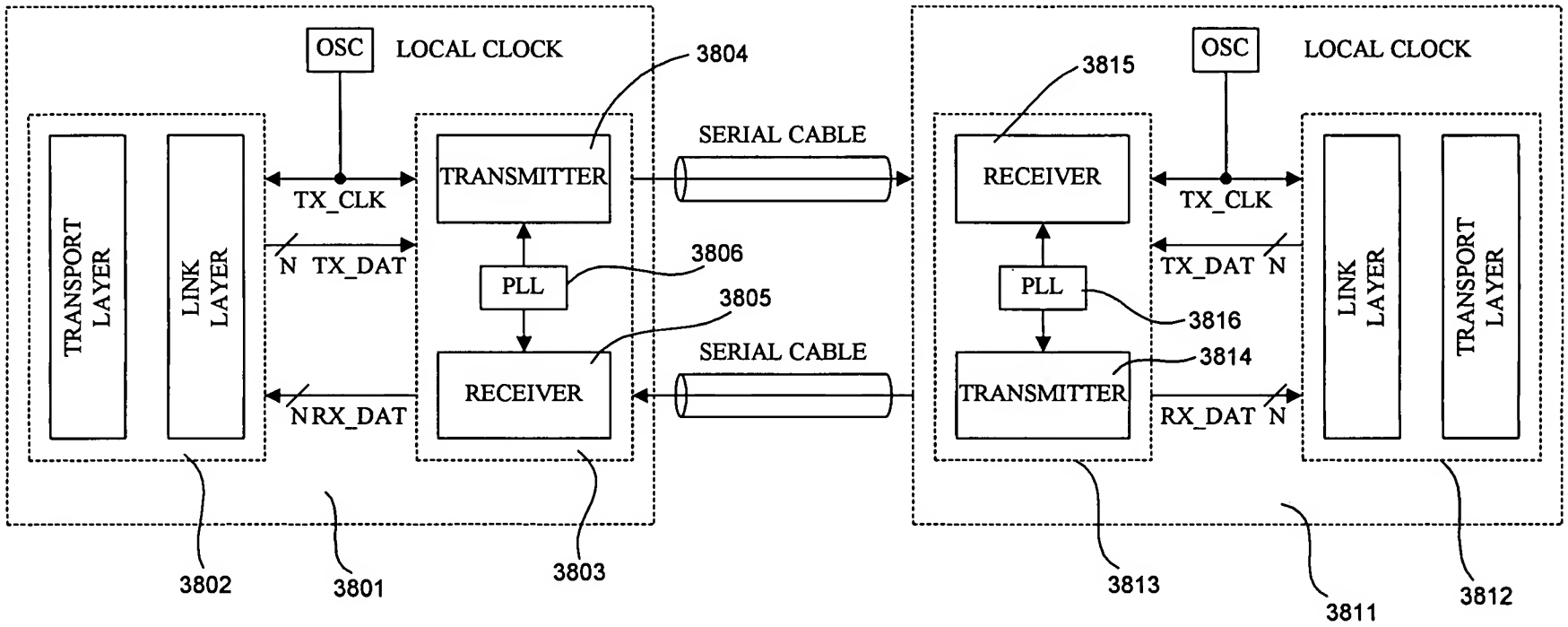



*Fig. 38A*

20STh0" h629E00T

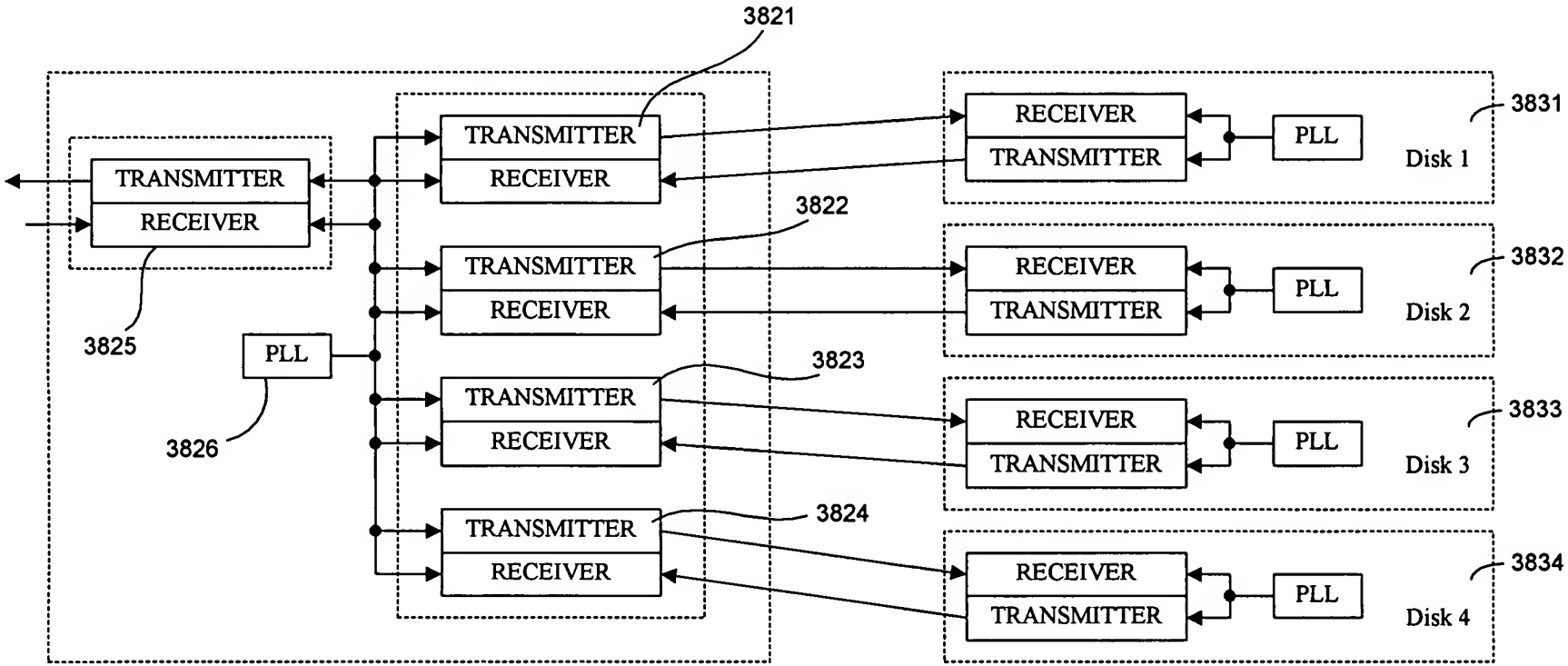



*Fig. 38B*

20STh0"h6Z9E00T

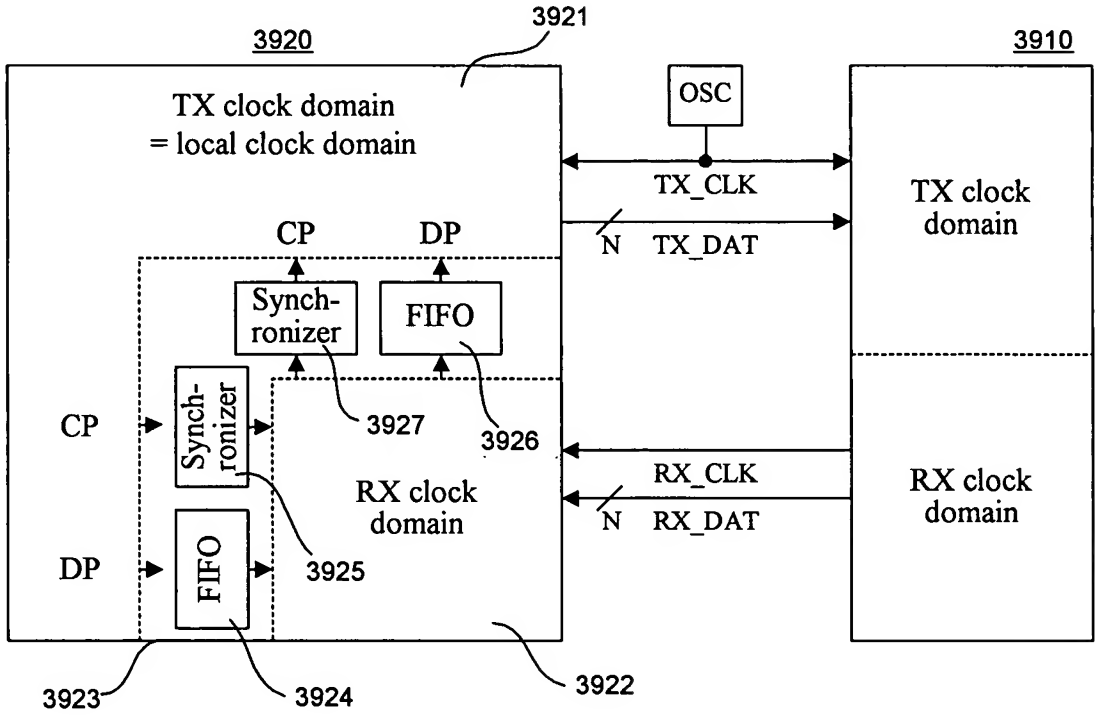



*Fig. 39A*

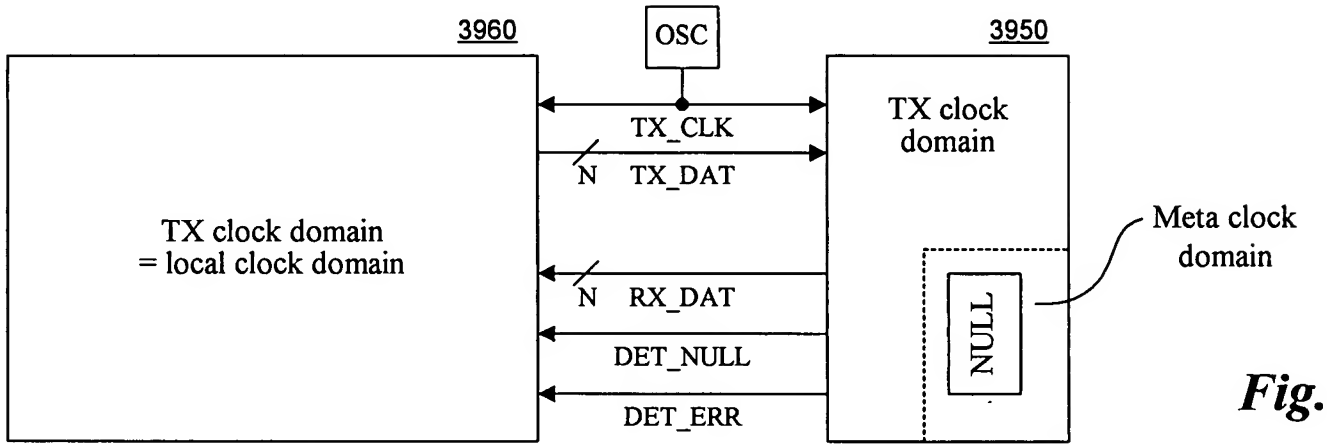



*Fig. 39B*

20STh0" h6Z9E00T

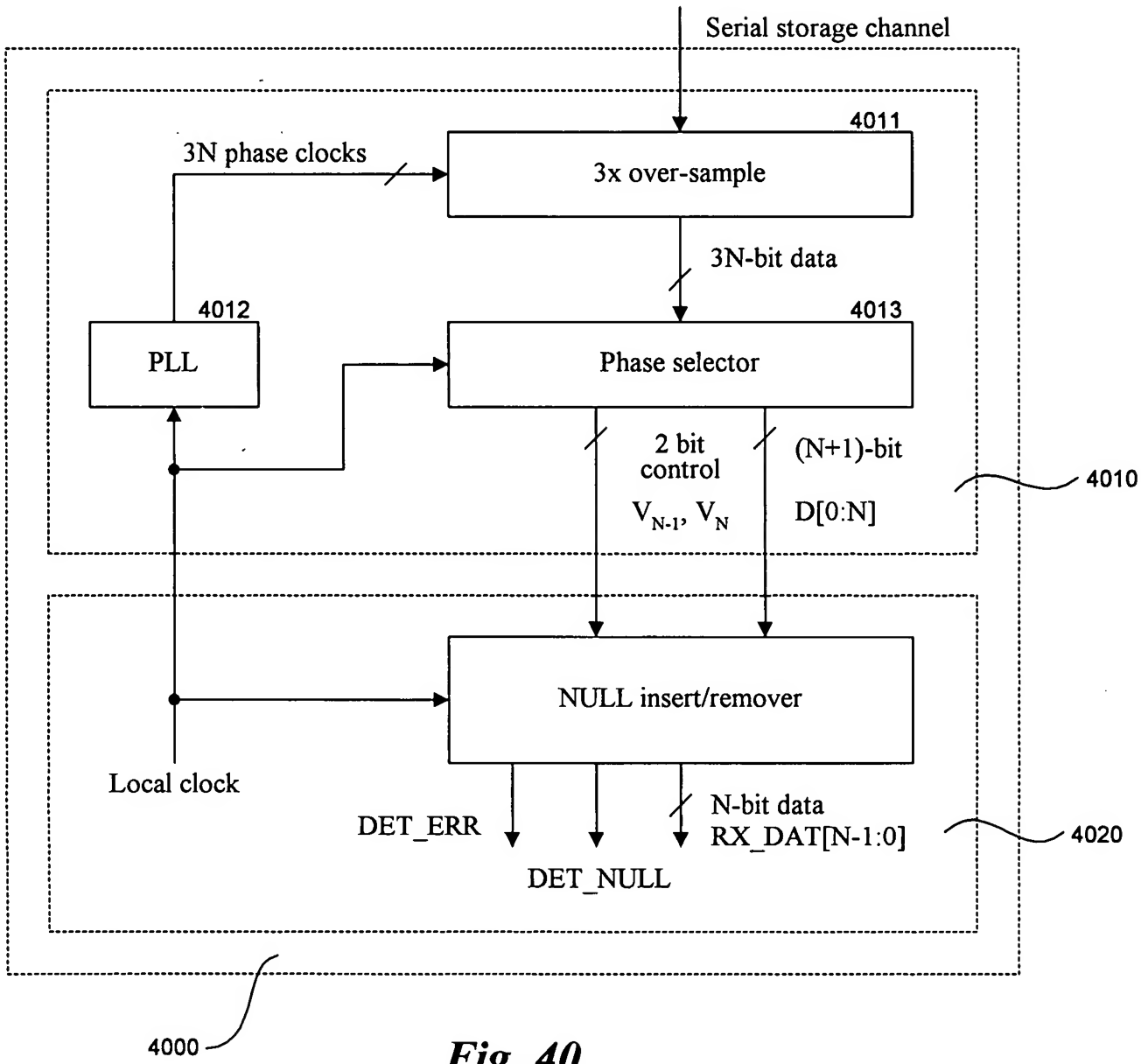



*Fig. 40*

2057h0* h6296007

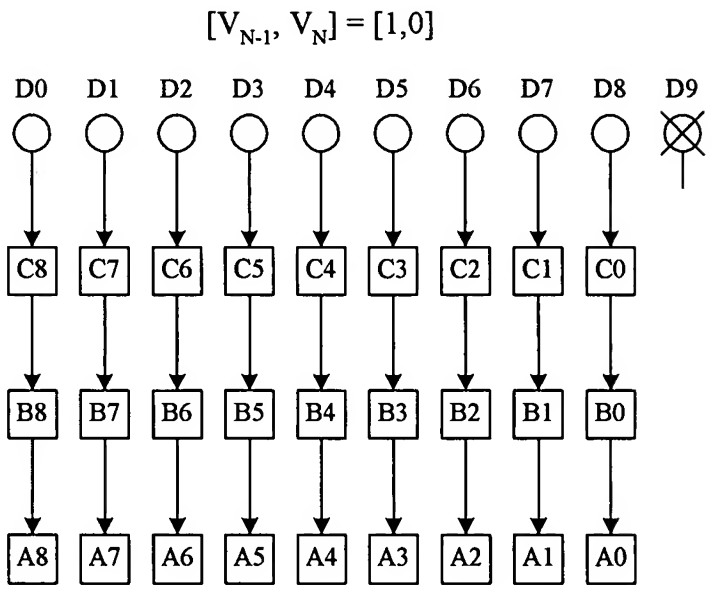



*Fig. 42A*

20STh0"h6Z9E00T

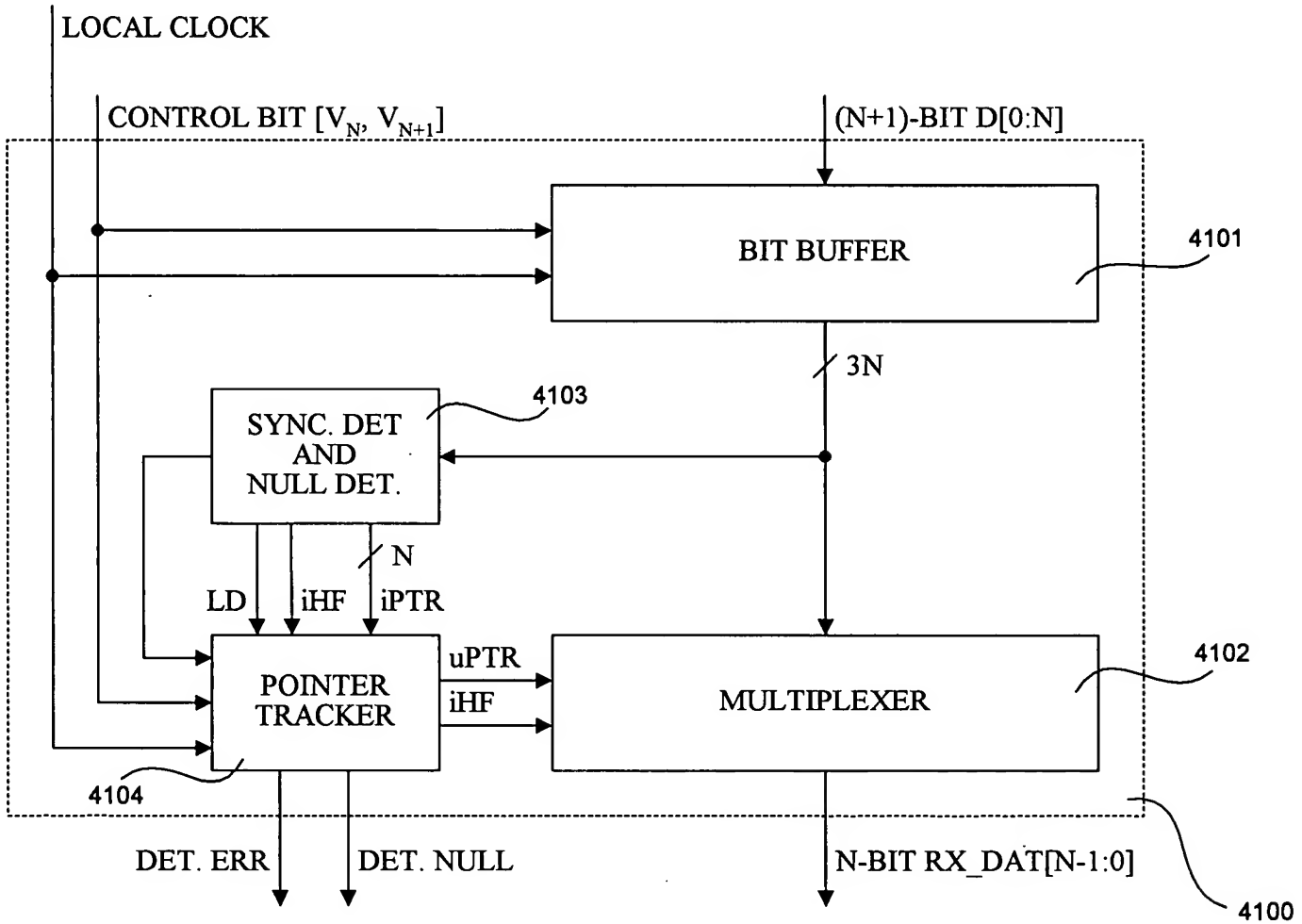



*Fig. 41*

20STH0" h6796007

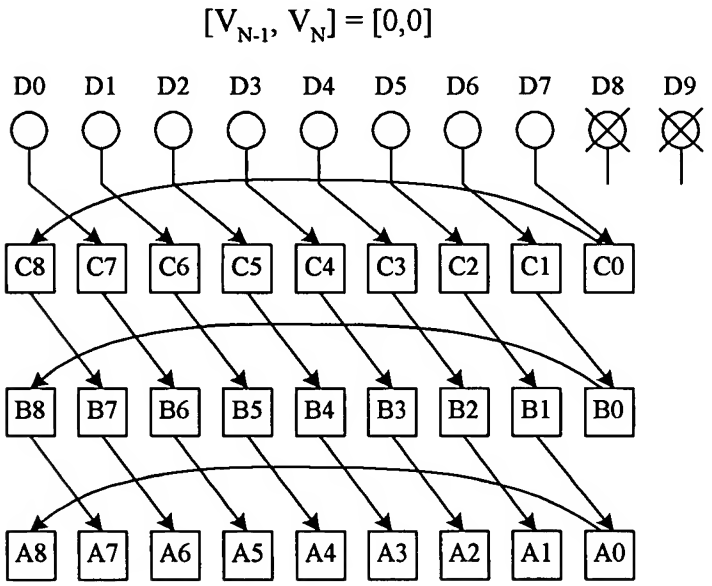



***Fig. 42B***

20STh0°h629E00T

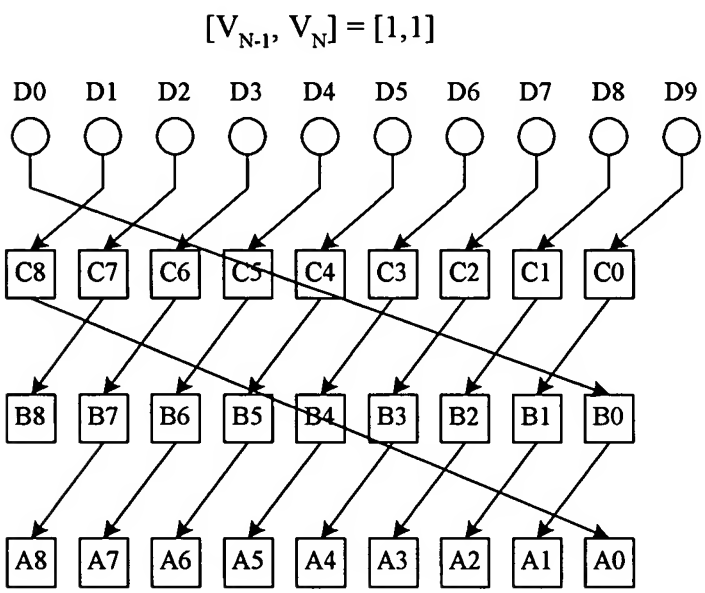



*Fig. 42C*

20STh0" h6Z9E00T

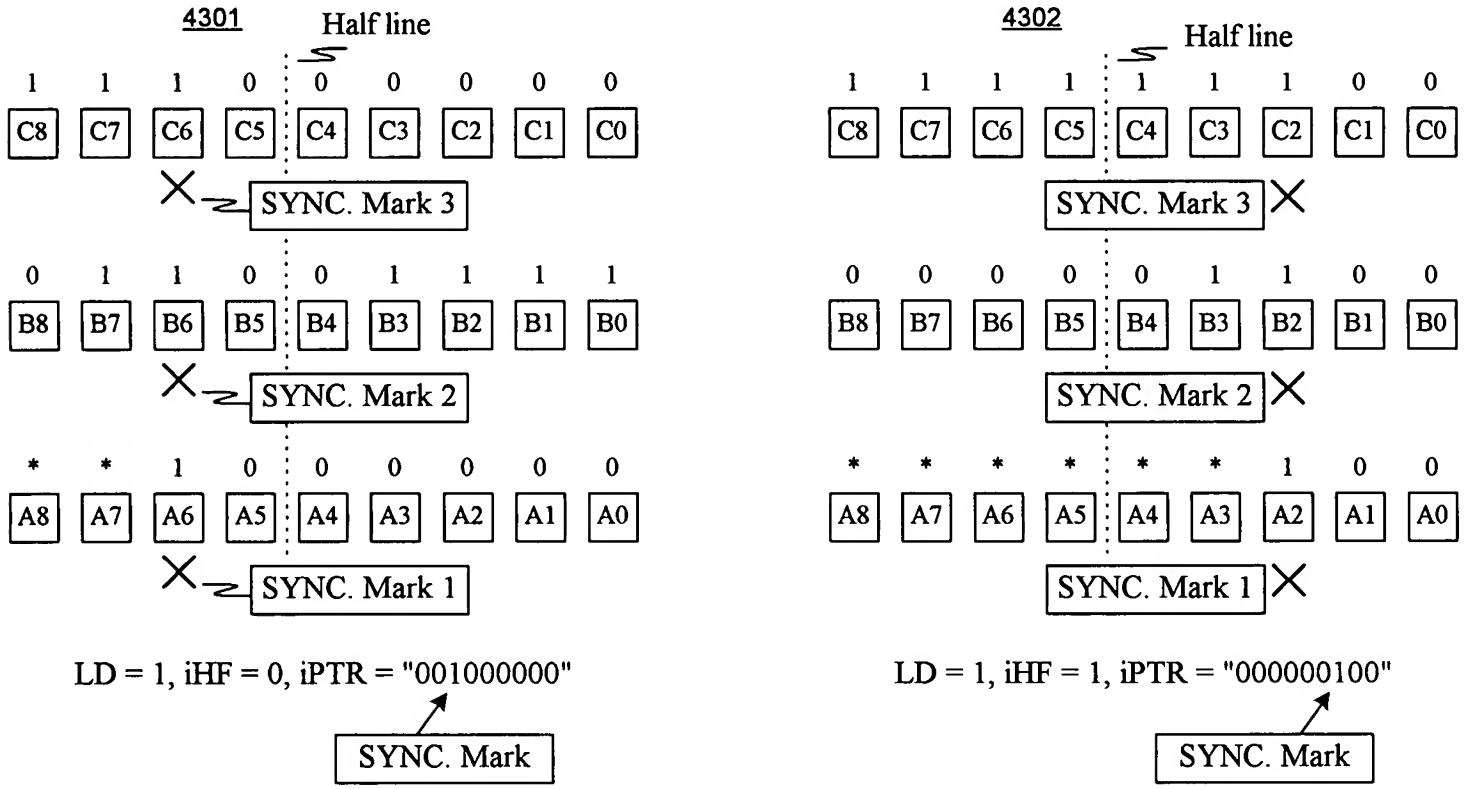
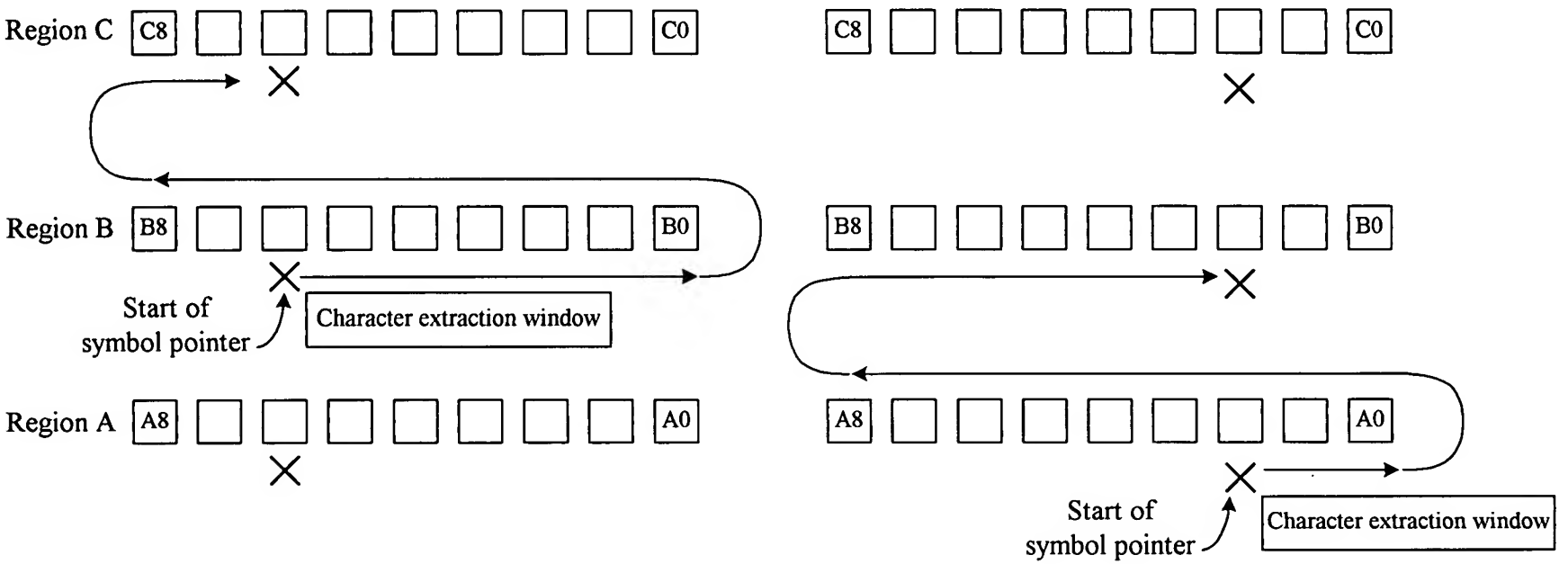



*Fig. 43*

20STh0"h629E00T




LD = 1, iHF = 0, iPTR = "001000000"

LD = 1, iHF = 1, iPTR = "000000100"

*Fig. 44*

20STh0"h6Z9E00T

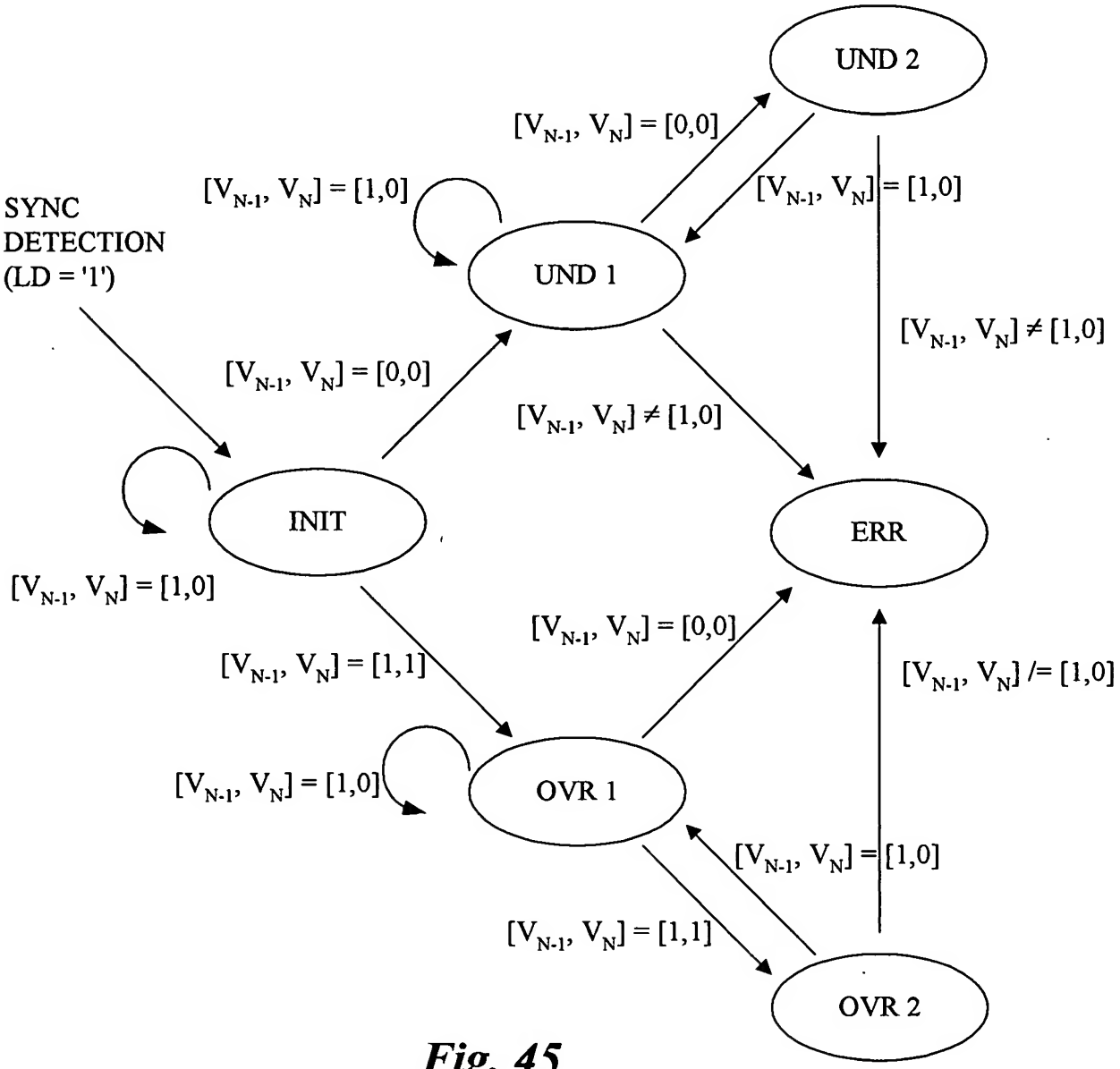



***Fig. 45***

20STh0"'h6Z9E00T

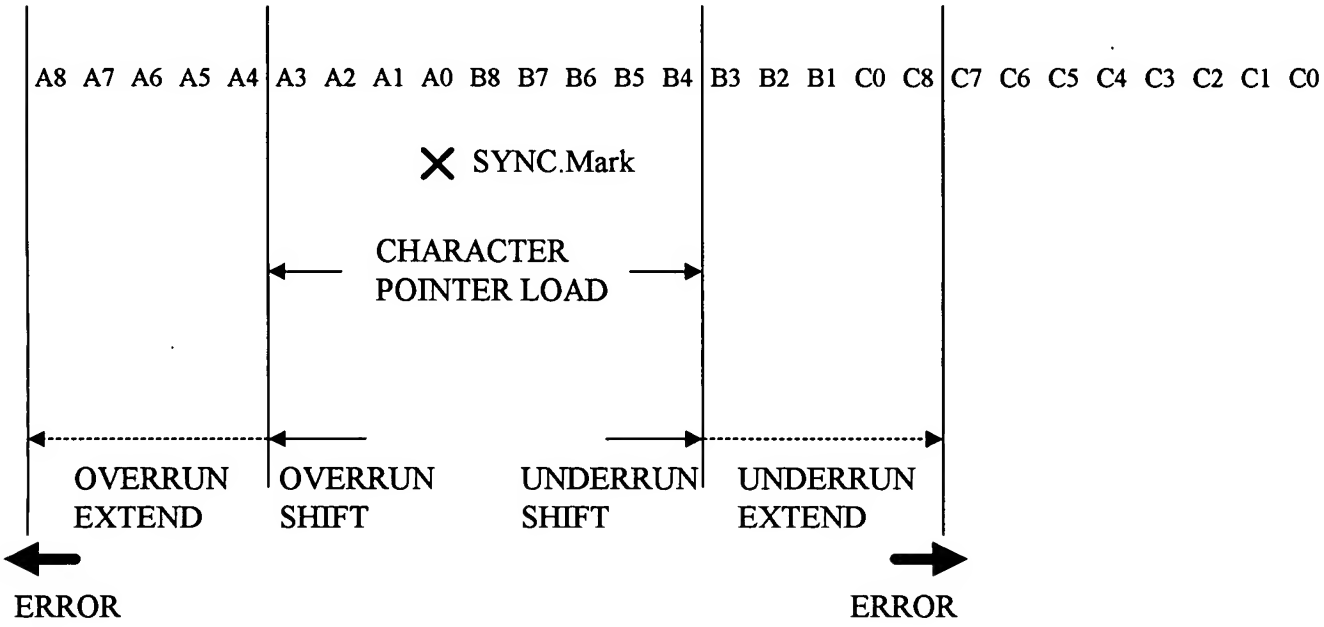



*Fig. 46*

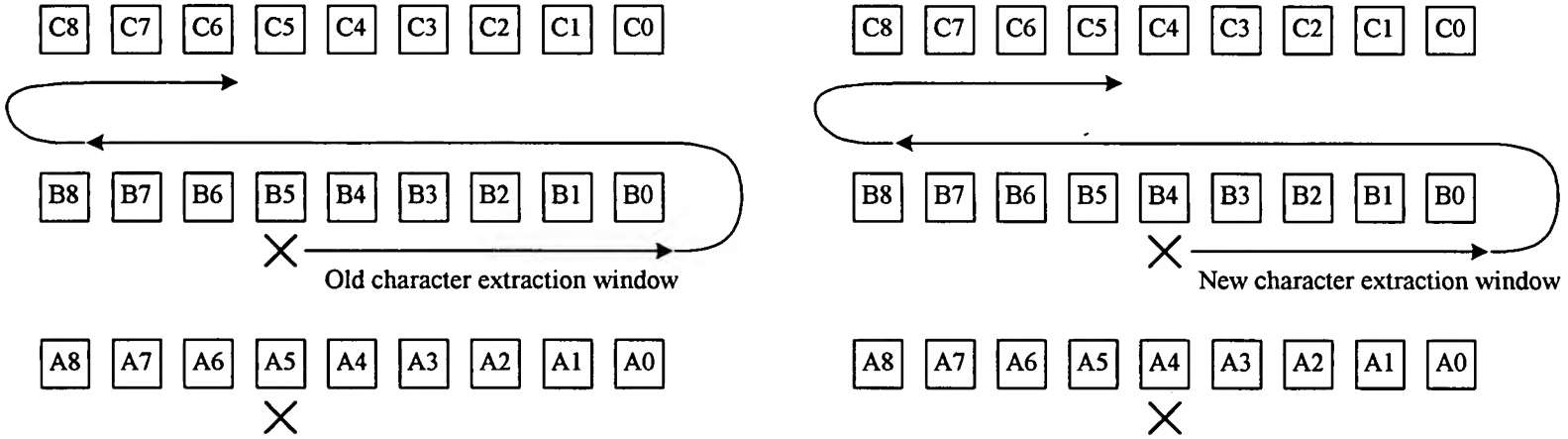



*Fig. 47A*

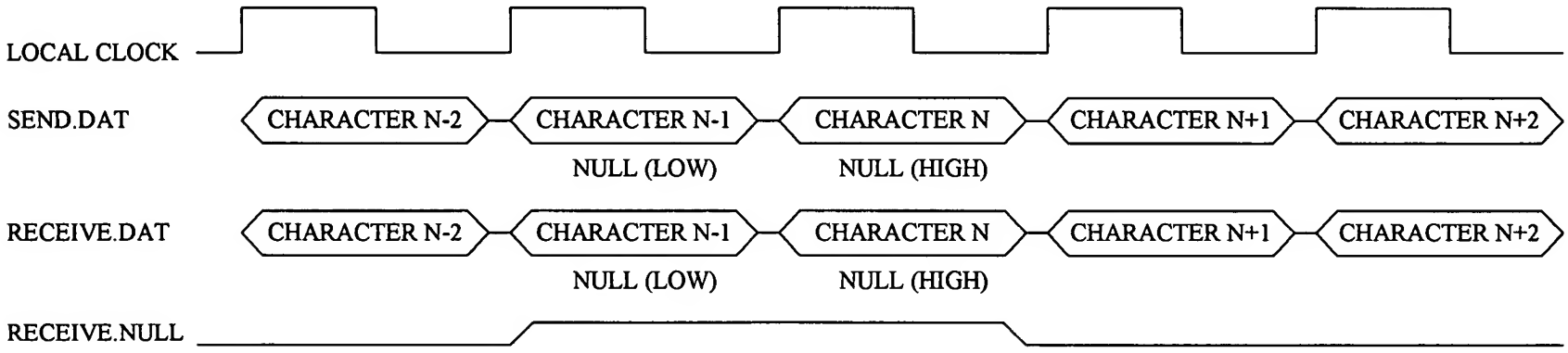



*Fig. 47B*

205tho" th629E00T

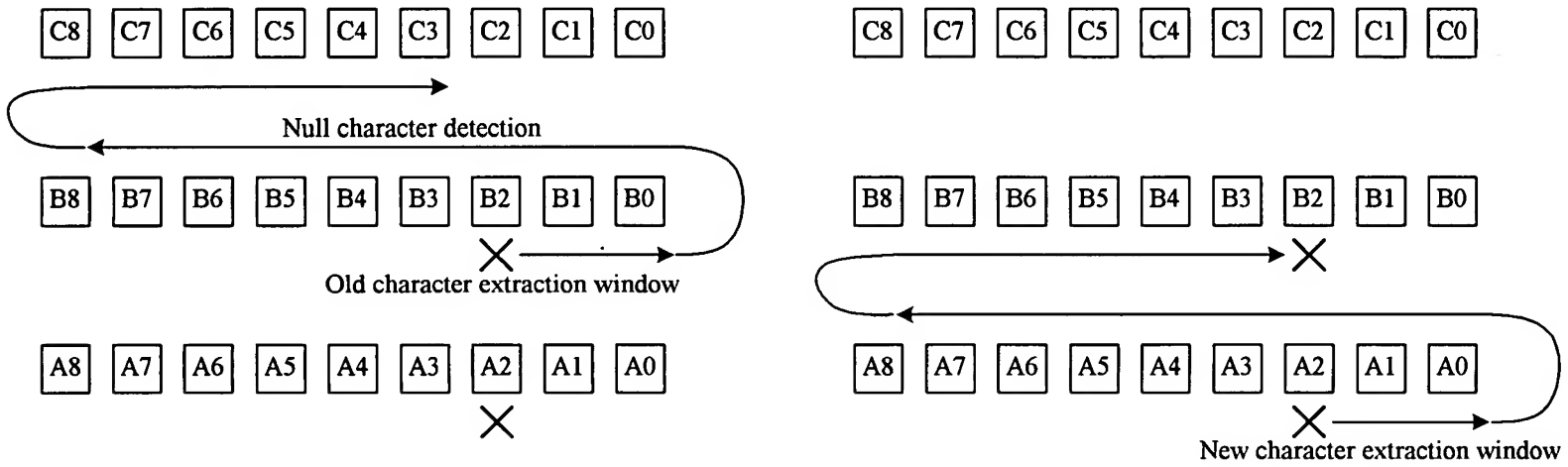



*Fig. 48A*

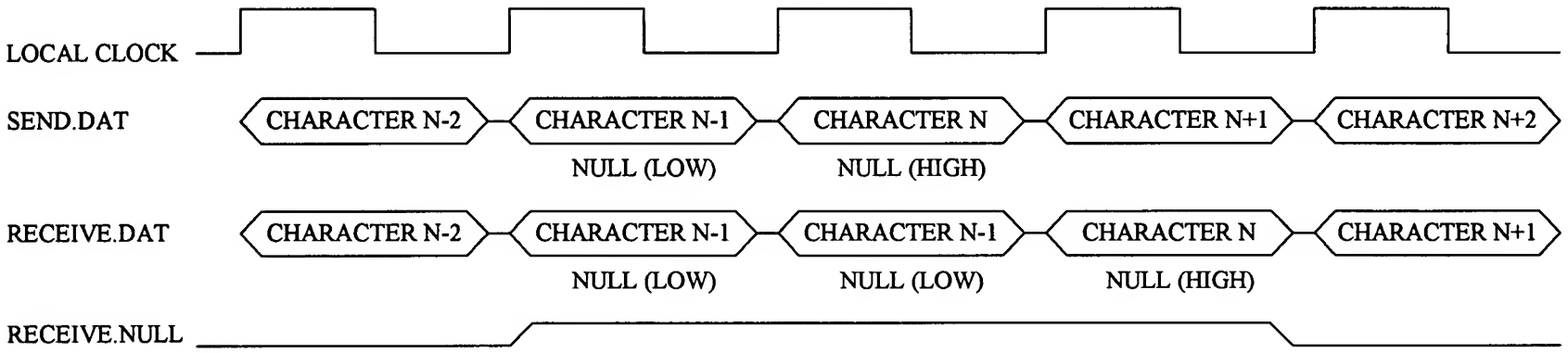



*Fig. 48B*

20STh0"h6Z9E00T

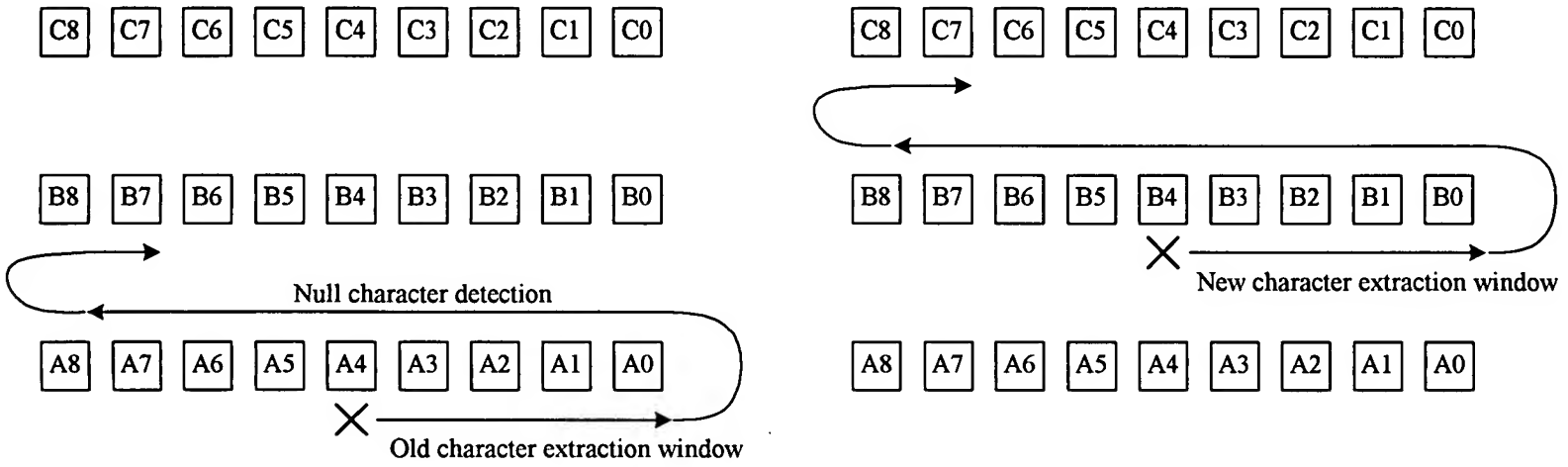



*Fig. 49A*

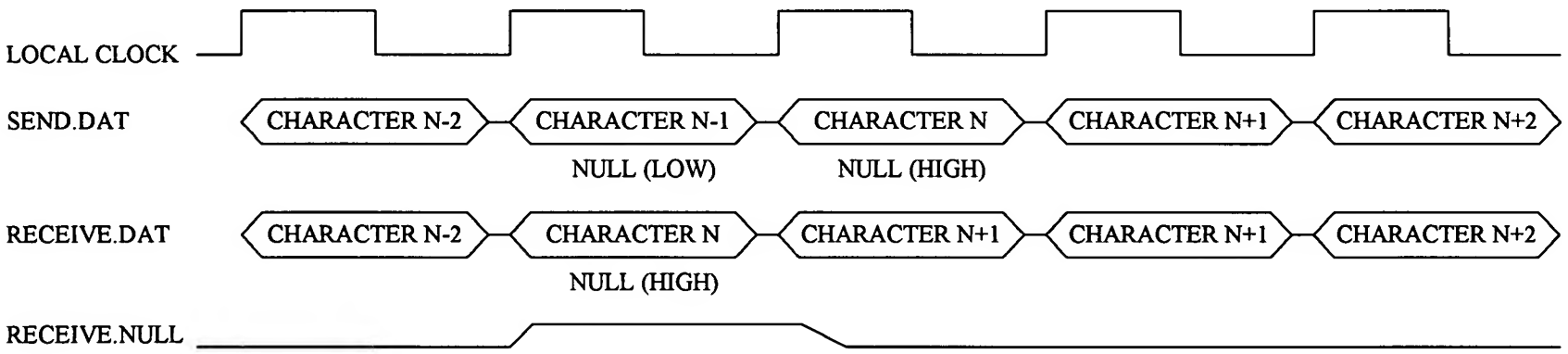



*Fig. 49B*